#23

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of	) Examiner: B. Edelman
James Prescott Curry	) Art Unit: 2153
	)
Serial No.: 09/449,237	) Atty. Docket No. 44884.12.1
	)
Filed: 24 Nov. 1999	)
	)
For: METHOD AND SYSTEM FOR PROVIDING ON-LINE WELLNESS) AND RETAIL ACCESS THROUGH) A DISTRIBUTED NETWORK	)

SENT VIA EXPRESS MAIL POST OFFICE TO ADDRESSEE

MAILING LABEL NO. EV 317158805 US

To: Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

10/15/2003 HAHMED1 00000001 09449237
02 FC:2402 165.00 OP

APPEAL BRIEF (37 C.F.R. § 1.192)

This is an appeal of the Examiner's final rejection of claims 81-93 and 95-97, issued on October 8, 2002. This brief is transmitted in triplicate.

REAL PARTY IN INTEREST.

This application remains the property of the inventor, James Prescott Curry, having the home address of 1504 Bay Rd. Apt C-2803, Miami Beach, FL 33139.

RELATED APPEALS AND INTERFERENCES.

None.

STATUS OF CLAIMS.

Claims 81-93 and 95-97 are pending in the case and have been finally rejected. Claims 1 – 80 and 94 have been cancelled without prejudice in previously filed amendments. A copy of the appealed claims is attached as Appendix A.

STATUS OF AMENDMENTS.

Since the Examiner issued the final rejection, Applicant filed its Notice of Appeal, and the Examiner filed an advisory action. The copy of the appealed claims attached as Appendix A.

SUMMARY OF INVENTION.

Physical fitness and wellness are worthy, if sometimes elusive goals for many people. Attaining these goals is often difficult, and any assistance provided by modern technology is

normally welcomed. Moreover, keeping track of workout routines and the progress made in those routines could be better served by improved use of modern technology.

The present invention provides methods and systems which allow a person to record wellness related data generally, and physical fitness data specifically, at the location most conveniently and closely located to the workout and weighing activities, specifically at the physical fitness center loci within physical access during workouts. The present invention provides a system and methods allowing a physical fitness workout activity to be recorded at a computer terminal or kiosk located at a synergistically related location, i.e., a physical fitness center. The invention also allows the user to later retrieve this stored information at another location, for example their home, over the Internet, to assess progress and anomalies with the perspective of rest and privacy.

The invention also provides a method providing an economic driving force for an organization to provide data entry terminals or kiosks for entering physical fitness data, located in a fitness center and coupled to the Internet. One such method allows a central service provider to distinguish between terminals that are sponsored by and located at a fitness center, and those terminals that are not sponsored and co-located. This may allow, for example, the user at a sponsored terminal to be directed toward buying products in the shop of that particular fitness center, or directed toward value added lessons or services provided by that particular fitness center. Non-sponsored terminals, for example, a home Internet connected computer, might not receive such directed advertising and prompting.

The invention also provides a method for providing wellness-related services including providing at least one control group for a user of a portal to an on-line wellness-related site. The

method also includes providing at least one control group, where the user is assigned to the control group automatically based on user attributes. This is particularly helpful because the method enables the providing of a user improvement plan which is based on the collective workout related attributes of the control group. Some methods adjust the improvement plan based on group result data. The relevance of this knowledge is high in this field of endeavor (i.e. exercise and health promotion psychology), particularly when properly utilized in the manner claimed by the invention. The method can also provide an alarm signal to a system administrator if the group plan needs to be adjusted and/or assign the user to a new control group based on result data for the user.

ISSUES PRESENTED.

Applicant believes the major issues are covered in the major groups I, II and III below. One major issue relates to the patentable weight to be given to the claim limitations in groups I and II.

I Whether claims 81 and 93 are obvious under §103(a) over Baker et al. (U.S. Patent No. 5,678,041) (copy attached as Appendix D).

I (A) Whether dependent claim 82 is patentable over Baker et al. in view of Szabo? (Appendix E).

I (B) Whether dependent claim 83 is patentable over Baker et al. in view of Szabo?

I (C) Whether dependent claim 84 is patentable over Baker et al. in view of Szabo?

I (D) Whether dependent claim 95 is patentable over Baker et al. in view of Szabo?

I (E) Whether dependent claim 97 is patentable over Baker et al. in view of Szabo?

I (F) Whether dependent claim 96 is patentable over Baker et al. in view of Szabo?

II Whether claim 85 is obvious under §103(a) over Baker et al. (U.S. Patent No. 5,678,041) (copy attached as Appendix D) in view of Szabo (U.S. Patent No. 5,954,640) (copy attached as Appendix E).

III Whether claim 86 is patentable over Baker et al. in view of Szabo?

III(A) Whether dependent claim 87 is patentable over Baker et al. in view of Szabo?

III(B) Whether dependent claim 90 is patentable over Baker et al. in view of Szabo?

III(C) Whether dependent claims 88, 89, 91 and 92 are patentable over Baker et al. in view of Szabo and further in view of Roth? (Appendix F).

GROUPING OF CLAIMS

The claims have been grouped into three groups I, II, and III dealing with the independent claims, with groups I and III subdivided into subgroups corresponding to the dependent claims. The claims in the groups and subgroups do not rise and fall together, as explained in the argument section, although subgroup III(C) claims do stand or fall together. The groupings are group I (claim 81 and 93); I(A) (claim 82); I(B) (claim 83); I(C) (claim 84); I(D) (claim 95); I(E) (claim 97); I(F) (claim 96); II (claim 85); III (claim 86); III(A) (claim 87); III(B) (claim 90); and III(C) (claims 88, 89, 91, and 92).

ARGUMENT.

The arguments have been grouped with the arguments relating to the independent claims

first, followed by the argument relating to the corresponding dependent claims.

INDEPENDENT CLAIMS

I Are claims 81 and 93 are obvious under §103(a) over Baker et al.

As noted in the Final Office Action for the present case (copy attached as Appendix C), the Examiner has rejected independent claims 81 and 93 under §103(a) based on Baker et al. (included in Appendix D). Claims charts summarizing the claims and Applicant's position are provided for the Board's convenience in Appendix B.

Claim 81 recites a method ... providing an on-line site that enables wellness-related databases to be accessed from at least one of a sponsored and non-sponsored portal, where the sponsored portal is at least in part sponsored by and located at a fitness center, where the non-sponsored portal can be accessed through the Internet and responding to a request based in part on whether the portal was sponsored. In the Final Rejection of Claim 81, the Examiner admits that Baker et al. "do not disclose using an online system for wellness services at a fitness center, wherein one of the multiple computers is a computer residing at the fitness center and is thus sponsored at the fitness center" Appendix C, page 3, paragraph 4). The Examiner further states that Claim 93 contains no further limitations over claim 81, except for providing a different level of services based at least in part on whether a request came from a sponsored portal located in a fitness center. The Examiner further admits that Baker et al. do not disclose a sponsored portal located in a health or fitness center. (Appendix C, page 4, paragraph 3).

The Examiner thus concedes that Baker et al. do not disclose the recited invention, and does not state that any combination of references teaches or suggests the invention recited in Claims 81 and 93.

Instead, the Examiner states that "the claim limitations that the database is a wellness-related database, and that one of the computers may be at a fitness center, and thus would be sponsored by the fitness center, are found only in the non-functional descriptive material and are not functionally related in the steps recited" (Appendix C, page 3, last paragraph). The Examiner further states that it would make no difference if the computers were located at a school, a health center, or a library. The Examiner finally states that the type of information and location does not functionally relate to the steps in the method claimed, and does not patentability distinguish the claimed invention. (Appendix C, page 4, second paragraph).

The Applicant stipulates that non-novel computer hardware, networking hardware, operating system software, networking software, and database packages could be used to implement the recited invention. The Applicant is not claiming any novelty in these individual items. Applicant respectfully disagrees with the Final Rejection on several grounds.

For the reasons given below, Applicant believes that the Examiner has read several limitations, and all of the business method limitations, out of the claims. In essence, the Examiner has added a technical effect requirement to U.S. Patent law that does not exist, while failing to properly apply the most relevant law.

The Examiner (though not so stating), may believe that the claim portal sponsorship, portal location, and data content limitations are merely business method limitations. Even if true, the claimed invention, which provides a means of storing and remotely retrieving health data obtained at an exercise facility, is patentable over the references cited by the Examiner. In *State Street Bank & Trust Co. v. Signature Financial Group, Inc.*, 139 F.3d 1368 (Fed. Cir. 1998), the United States Court of Appeals stated that "Since the 1952 Patent Act, business

methods have been, and should have been, subject to the same legal requirements for patentability as applied to any other process or method." *Ibid.* In addition, the court elaborated on how such claims should be analyzed for patentability. "The mere fact that a claimed invention involves inputting numbers, calculating numbers, outputting numbers, and storing numbers, in and of itself, would not render it nonstatutory subject matter, unless, or of course, its operation does not produce a "useful and concrete tangible result." *Ibid.* The present invention clearly satisfies this standard by facilitating the storage of fitness data at the site where it is created, and then enabling the data to be retrieved remotely at the convenience of the individual user. The patentability of "business method" inventions was further reinforced in *AT&T Corp. v. Excel Communications Inc.*, 172 F.3d 1352 (Fed. Cir. 1999), in which the court reaffirmed its earlier holding that a useful business method was patentable subject matter, stating that "the sea-changes in both law and technology stand as a testament to the ability of law to adapt to new and innovative concepts, while remaining true to its basic principles." *Ibid.*

Thus even business method limitations are not to be disregarded simply because they are related to business aspects. Applicant submits that the content of the database, the sponsorship of the portals, and the location of the portals are claim limitations and are functionally related to each other.

The Examiner asserts that Applicant has no allowable patent claims because Applicant's claim limitations are merely "descriptive" rather than "functional." Applicant has repeatedly and expressly stated that any such terms are meant to be both descriptive and functional in the claimed invention. There is simply no basis for the Examiner's rationale. The proper claim

construction for these claims includes extensive specification and prosecution history explanation of the meaning and scope of the words chosen by Applicant. Applicant has consistently, clearly and unambiguously stated the functional meaning and scope of the language of the claims. The Examiner's failure to accept these expressions of claim language meaning, which is incidentally well within the normal usage of such language, does not reconcile with current law. One skilled in the art of these claims would readily recognize the functionally limiting nature of the works at issue, particularly in view of the specification and prosecution history. Not only are the functional (and descriptive) limitations adequate to notify the public of the scope of the claims' exclusivity, but the Applicant has also claim features and functions which enable technical solutions to problems unique to the relevant fields of wellness and exercise psychology.

The Applicant has clearly disavowed claim scope during prosecution and by his careful use of words in the claims. There is adequate functional limitations in the claims, as applied to the relevant fields of application to support non-obviousness and novelty over the cited references.

First, in a business context, the sponsorship of a portal in a fitness center and the location of the portal in the fitness center are clearly limitations, as a portal not being so sponsored and located would not literally infringe a claim containing this language. The words clearly limit the claim. The notice function of the patent is well and clearly served by these words that limit the patent claims.

Second, in a business context, the sponsorship of a portal in a fitness center and the location of the portal in the fitness center are clearly functionally related to each other, as the

fitness center is more likely to pay for a portal located in the center, where there are members, and where there are workout facilities to generate data to enter into the portal. A fitness center is also more likely to pay for a portal to inform a member about revenue generating activities and products available at the fitness center. An unrelated party is less likely to sponsor a portal to sell products for, or provide information about, the fitness center. Thus the "placing in communication" step, the "processing step", and the "responding" step of claim 81 are indeed functionally related to the location and sponsorship of the portal.

Third, the content of the database is functionally related to the steps, as the content of the wellness related data is likely to be generated at the fitness center. For example, providing a portal for entry of workout or weight information makes perfect sense at a fitness center, where such data is generated, and makes no sense at the library location cited by the Examiner. If there was not functional relationship, then entering workout data at a library would make as much sense as at a fitness center. In another example, the general health or wellness content is functionally related to the location as people at a fitness center are more likely to be interested in and immediately focused on such content (which may also be sold as products at the fitness center) than people at large, for example, at libraries. The portal users in a fitness center are thus targeted, segmented users functionally related to the wellness content. Thus the wellness content of the database is functionally related to the "providing" step, the "placing in communication" step, and the "responding" step.

In the Final Office Action, pages 11-13, the Examiner concisely states his position and makes further arguments. The Examiner states that placing terminals in schools or businesses, and varying the level of access depending on the location of the terminals, discloses the

sponsorship limitation of the recited invention. Again, Applicant submits that the Examiner is examining the invention after reading out all relevant limitations from the claims.

II Whether claim 85 is obvious under §103(a) over Baker et al. in view of Szabo

As noted in the Final Office Action for the present case (copy attached as Appendix C), the Examiner has rejected independent claim 85 under §103(a) based on Baker et al. (included in Appendix D) in view of Szabo (U.S. Patent No. 5,954,640) (included in Appendix E) (Appendix C, page 6, second paragraph).

Claim 85 recites that users are able to enter fitness related data selected from the group consisting of workout plans, workout goals, weight training plans, weight training weights, and weight training repetitions at the fitness center and view the fitness data from non-sponsored portals. In fact, the Examiner did not base his rejection on a combination of Baker et al. and Szabo. Rather the Examiner admitted that Baker et al. did not disclose the elements of claim 85 with respect to Claim 81, discussed above. (Appendix C, page 3, paragraph 4).

Instead, the Examiner rejected claim 85 "for the same reasons as stated above" (with respect to Claims 81 and 82). (Appendix C, page 6, second paragraph). The Examiner stated that the recited Markush group of fitness data was non-functional and descriptive, and that it did not add to the functional operation of the claimed invention in fact the Examiner stated that data could be any data and still perform the same function. (Appendix C, page 6, paragraph 2).

For the sake of brevity, as the Examiner referred to his previous argument for claim 81, so shall we. Please refer to the argument made above with respect to claims 81 and 93. Applicant submits that fitness related data selected from the group consisting of workout plans,

workout goals, weight training plans, weight training weights, and weight training repetitions, is functionally related to the location of the portal in a fitness center, which is the location of the workouts and the data generated from the workouts. The Examiner states that the data could be any data (including "business plan data") and that "the invention would still perform the same function" (Appendix C, page 6, second paragraph). Applicant submits that the business plan data cited by the Examiner would not perform the same function to a fitness center member in fitness center as being able to enter the data from their workout.

The Examiner has focused on the computer hardware and software elements, and read the other limitations out of the claim. These other, fitness-related data limitations are functionally related to the "providing" and "placing in communication" steps of claim 85. The examiner has not properly recognized these limitations and linking relationships and appears not to recognize the admonitions to do so as stated by the court of appeals for the Federal Circuit. There is no "technical effect" requirement for patentability in U.S. patent law.

The Examiner cited *In re Lowry* (32 F.3d 1579, 32 USPQ2d 1031 (Fed. Cir. 1994) for the proposition that "descriptive" material will not distinguish a claimed invention from the prior art in terms of patentability (Appendix C, page 3, last paragraph, through page 4, first paragraph). The *Lowry* court stated that "the burden of establishing the absence of a novel, nonobvious functional relationship rests with the PTO. 'If examination at the initial stage does not produce a prima facie case of unpatentability, then without more the applicant is entitled to grant of the patent' " (citing *In re Oetiker*, 977 F.2d 1443, 1445, 24 USPQ2d 1443, 1444 (Fed. Cir. 1992) (emphasis added). Applicant submits that, given the functional relationships described between the claim limitations above, the Examiner has therefore not established a prima facie case of

unpatentability.

Applicant appreciates the difficulty in prior art searching in inventive areas such as this. However, a proper rejection under 35 U.S.C. 103(a) should be based on a reference or set of references that teach or suggest the claimed invention. The rejection should not be based on dismissing the limitations of major portions of the claimed invention, then citing references that do not teach the claimed invention. In the present appeal, the rejection should not be based on deleting the non-computer hardware-software limitations, even though functionally inter-related, and then rejecting what remains of the claims over references that do not teach the originally claimed invention.

III Whether claim 86 is patentable over Baker et al. in view of Szabo?

The Examiner rejected Claim 86 under 35 U.S.C. 103(a) as unpatentable over Baker et al., in view of Szabo (Appendix C, page 4, last paragraph). Claim 86 recites a method of providing well-ness related services comprising .... providing at least one control group and automatically assigning the user to one of control groups based on user attributes. The Examiner quotes Szabo "the models themselves may be adaptive based on the experience of individual users or groups ... neural network technology and other adaptive paradigms may be employed to dynamically improve the models through use and feedback" (Appendix C, page 5, last paragraph) citing Szabo, column 9, line 66-column 10, line 9. Applicant respectfully disagrees. Szabo discusses adaptive models for users already in groups, not automatically assigning users to groups based on user attributes.

DEPENDENT CLAIMS

I(A) Whether dependent claim 82 is patentable over Baker et al. in view of Szabo?

Claim 82 depends from claim 81, and for all the reasons discussed above with respect to claim 81 is patentable as well. The Examiner rejected claim 82 over Baker et al. in view of Szabo (Appendix C, page 4, last paragraph through page 5, first paragraph). Claim 82 further recites entering fitness related data into the data base and providing access to the user to the fitness data through the non-sponsored portal through the Internet. The Examiner cites Szabo (col. 6, lines 5-9; col. 3, lines 56-61). Szabo teaches a point of sale dispensing machine for dispensing nutritional supplementation and advice. See col. 5, line 65 - col. 6, line 11. The Examiner also cites Baker et al. Fig. 1 and Szabo (col. 4, lines 16-23), where Baker et al. shows multiple computers connected to a network, and where Szabo discusses using multiple computers and the Internet to enter data. Neither reference teaches entering fitness related data into the data base and providing access to the user to the fitness data through the non-sponsored portal through the Internet.

I (B) Whether dependent claim 83 is patentable over Baker et al. in view of Szabo?

Claim 83 depends from claims 81 and 82, and for all the reasons discussed above with respect to claims 82 and 83, is patentable as well. The Examiner rejected claim 83 over Baker et al. in view of Szabo (Appendix C, page 5, last paragraph). Claim 83 further recites automatically assigning the user to a control group based on user attributes. The Examiner quotes Szabo "the models themselves may be adaptive based on the experience of individual users or groups ...

neural network technology and other adaptive paradigms may be employed to dynamically improve the models through use and feedback." (Appendix C, page 5, last paragraph, citing Szabo, column 9, line 66-column 10, line 9). Applicant respectfully disagrees. Szabo discusses adaptive models for users already in groups, not automatically assigning users to groups based on user attributes.

I(C) Whether dependent claim 84 is patentable over Baker et al. in view of Szabo?

Claim 84 depends from claims 81, 82 and 83 and for all the reasons discussed above with respect to claims 81, 82, and 83, is patentable as well. The Examiner rejected claim 84 over Baker et al. in view of Szabo (Appendix C, page 6, first paragraph, citing Szabo, col. 10, lines 1-34). Claim 84 further recites providing fitness advice and goals to the control group, wherein the advice and goals are at least in part a result of the group result data. The cited lines in Szabo discuss nutritional supplements. Szabo discusses the possibility of adding exercise (Col. 10, lines 35-41), but this would be "difficult to effect" and "of limited benefit" and "likely to be rejected or ignored." Szabo thus hardly teaches providing fitness advice and goals to the control group, wherein the advice and goals are at least in part a result of the group result data.

I (D) Whether dependent claim 95 is patentable over Baker et al. in view of Szabo?

Claim 95 depends from claim 81, 82, and 83, and for all the reasons discussed above with respect to claims 81, 82, and 83, is patentable as well. The Examiner rejected claim 95 over

Baker et al. in view of Szabo (Appendix C, page 8, third paragraph, citing Szabo, col. 10, lines 1-34). Claim 95 further recites creating practical guidelines and advice for the control group, and wherein the services providing step comprises providing a user improvement plan for the user, the user improvement plan is selected to be similar to the practical guidelines and advice for the control group. The cited lines in Szabo discuss nutritional supplements. Szabo discusses the possibility of adding exercise (Col. 10, lines 35-41), but this would be "difficult to effect" and "of limited benefit" and "likely to be rejected or ignored." Szabo thus hardly teaches providing fitness advice and goals to the control group, wherein the advice and goals are at least in part a result of the group result data, much less user improvement plans add to this fitness advice.

I (E) Whether dependent claim 97 is patentable over Baker et al. in view of Szabo?

Claim 97 depends from claim 81, 82, 83, and 95, and for all the reasons discussed above with respect to claims 81, 82, 83 and 95 is patentable as well. The Examiner rejected claim 97 over Baker et al. in view of Szabo (Appendix C, page 8, fourth paragraph, citing Szabo, col. 10, lines 14-34). Claim 97 further recites assigning the user to a new control group based on the stored result data for the user. The cited lines in Szabo discuss nutritional supplements. Szabo discusses the possibility of adding exercise (Col. 10, lines 35-41), but this would be "difficult to effect" and "of limited benefit" and "likely to be rejected or ignored." Szabo thus hardly teaches providing fitness advice and goals to the control group, wherein the advice and goals are at least in part a result of the group result data, much less teaching assigning a user to a new control group based on stored result data for the user.

I (F) Whether dependent claim 96 is patentable over Baker et al. in view of Szabo?

The Examiner rejected claim 96 (Appendix C, page 8, second paragraph), citing Szabo (col. 10, lines 1-34). Claim 96 depends from claim 81, 82, 83, and 95, and thus requires fitness-related data entered by the user through the sponsored portal at the fitness center, entering the fitness data for the user into the database, and providing access to the user to the fitness data through the non-sponsored portal through the Internet. Claim 96 further recites adjusting the user improvement plan for each user in the authorized user's control group based on the stored group result data. The cited lines in Szabo discuss nutritional supplements. Szabo discusses the possibility of adding exercise (Col. 10, lines 35-41), but this would be "difficult to effect" and "of limited benefit" and "likely to be rejected or ignored." Szabo thus hardly teaches the accumulated limitations of claim 96.

III(A) Whether dependent claim 87 is patentable over Baker et al. in view of Szabo?

Claim 87 depends from claim 86, and for the reasons above discussed with respect to claim 86, is patentable as well. Claim 87 further recites providing information or goods to the user based upon the control group to which the user has been assigned. The Examiner rejected claim 86 (Appendix C, page 8, first paragraph), citing Szabo (Col. 9, line 66-col. 10, line 9). In this section, Szabo discusses adaptive health models based on the experiences of individual users or groups of users. Szabo does not disclose providing information or goods to the user based upon the control group to which the user has been assigned, where the assigning was done

automatically as required by claim 86.

III(B) Whether dependent claim 90 is patentable over Baker et al. in view of Szabo?

The Examiner rejected claim 90 (Appendix C, page 8, second paragraph), citing Szabo (col. 10, lines 1-34). Claim 90 further recites adjusting the user improvement plan for each user in the authorized user's control group based on the stored group result data. Claim 90 depends from claim 86, and for the reasons above discussed with respect to claim 86, is patentable as well over Baker in view of Szabo.

III(C) Whether dependent claims 88, 89, 91 and 92 are patentable over Baker et al. in view of Szabo and further in view of Roth?

Claims 88, 89, 91, and 92 depend from claim 86, contain all the limitations of claim 86. Claim 86, as discussed above, includes assigning a user to a control group, wherein the assigning is done automatically based on user attributes. Baker et al., Szabo, and Roth, either alone or in combination, to not disclose this limitation. Claims 88, 89, 91, and 92 are therefore patentable over Baker et al. in view of Szabo and further in view of Roth.

CONCLUSION

The Examiner cited *In re Lowry* for the proposition that "descriptive" material will not distinguish a claimed invention from the prior art in terms of patentability The *Lowry* court stated that "the burden of establishing the absence of a novel, nonobvious functional relationship

rests with the PTO. If examination at the initial stage does not produce a prima facie case of unpatentability, then without more the applicant is entitled to grant of the patent" (emphasis added). Applicant submits that, given the functional relationships described between the claim limitations above, the Examiner has not established a prima facie case of unpatentability.

Applicant appreciates the difficulty in prior art searching in inventive areas such as this. However, a proper rejection under 35 U.S.C. 103(a) should be based on a reference or set of references that teach or suggest the claimed invention. The rejection should not be based on dismissing the limitations of major portions of the claimed invention, then citing references that do not teach the claimed invention. In the present appeal, the rejection should not be based on deleting the non-computer hardware-software limitations, even though functionally inter-related, and then rejecting what remains of the claims over references that do not teach the originally claimed invention.

The present invention, filed four years ago, provided methods for entering fitness data at a fitness center and viewing them at home. The invention also provided fitness centers financial reasons synergistically related to the content, location, use and properties of the kiosks to provide kiosks for their members. The prior art provided by the Examiner did not provide this. This has further resulted in a curtailment of patent rights to the Applicant and is an improper application of the law. Rather a failure to identify and apply specific prior art should lead to allowance rather than Final Rejections of these claims.

Reversal of the rejection of claims 81-93 and 95-97 is believed warranted and is solicited.

Respectfully submitted,

Dated: Oct. 8, 2003

Craig F. Taylor
Reg. No. 40,199
FREDRIKSON & BYRON, P.A.
Cust. No. **022859**
4000 Pillsbury Center
200 South Sixth Street
Minneapolis MN 55402-1425
(612) 492-7000

APPENDIX A

(Claims involved in the appeal)

81. A method of providing wellness-related services, including at least one of wellness, health, or fitness services through a publicly accessible distributed network to authorized users using authorized portals, comprising:

providing an online site that enables wellness-related databases to be accessed from at least one of a sponsored and a non-sponsored portal;

placing in communication at least one of a sponsored and non-sponsored portal to the online site through the publicly accessible distributed network wherein the publicly accessible distributed network includes the Internet, wherein the sponsored portal is at least in part sponsored by and located at, a fitness center, and wherein at least one of the non-sponsored portals accesses the on-line site through the Internet;

receiving a request at the online site requesting access to the wellness-related databases;

processing the request at the online site to determine whether the portal was sponsored and whether the request was received from an authorized user; and

responding to the request based in part on whether the portal was sponsored and whether the user is authorized.

82. A method as in Claim 81, wherein the method further comprises obtaining fitness-

related data from the user through the sponsored portal at the fitness center, entering the fitness data for the user into the database, and providing access to the user to the fitness data through the non-sponsored portal through the Internet.

83. A method as in Claim 82, further comprising automatically assigning the user to a control group based on user attributes.

84. A method as in Claim 83, further comprising providing fitness advice and goals to the control group, wherein the advice and goals are at least in part a result of the group result data.

85. A method of providing wellness-related services, including at least one of wellness, health, or fitness services through a publicly accessible distributed network to authorized users using authorized portals, comprising:

providing an online site that enables wellness-related databases to be accessed from at least one of a sponsored and a non-sponsored portal; and

placing in communication at least one of a sponsored and non-sponsored portal to the online site through a publicly available distributed network;

wherein at least one of the sponsored portals includes a computer display located in a fitness center, wherein the method further comprises providing access to the authorized users at both the fitness center sponsored portal and the non-sponsored portals, wherein the authorized users are able to enter fitness-related data selected from the group consisting of workout plans,

workout goals, weight training plans, weight training weights and weight training repetitions at the fitness center and view the fitness data from the non-sponsored portals.

86. A method of providing wellness-related services, including at least one of wellness, health or fitness services to an authorized user through a distributed communications network, comprising:

identifying a portal with a portal identifier;

storing the portal identifier associated with the portal in a database;

receiving a request from the portal by an online wellness-related site;

processing the request at a controller to determine whether the request was from the portal;

assigning an access code to the user, the access code defining a level of wellness-related services available to the user;

providing services to the user through the distributed network that corresponds to the user's access code;

providing at least one control group, wherein each control group includes at least one authorized user; and

assigning the user to one of the control groups, wherein the assigning is done automatically based on user attributes.

87. A method as in Claim 86, wherein the services providing step comprises providing information or goods to the user based upon the control group to which the user has been assigned.

88. A method as in Claim 86, further comprising the step of creating practical workout guidelines and workout advice for the control group, wherein the services providing step comprises providing a user improvement plan for the user, and the user improvement plan is selected to be related to the practical guidelines and advice for the control group.

89. A method as in Claim 88, wherein the user improvement plan is at least in part based on the collective workout related attributes of the control group.

90. A method as in Claim 86, wherein each control group includes group result data, the method further comprising the steps of:

providing the result data to the portal;

storing the result data to the group result data for the authorized user's control group; and

adjusting the user improvement plan for each user in the authorized user's control group based on the stored group result data.

91. A method as in Claim 88, further comprising the step of providing an alarm signal

to a system administrator if the user improvement plan for the users in the control group needs to be adjusted.

92. The method as in Claim 88, further comprising:

storing result data for the authorized user; and

assigning the user to a new control group based on the stored result data for the user.

93. A method of providing wellness-related services, including at least one of wellness, health, or fitness services through a distributed communications network, wherein the network is coupled to an on-line wellness related site and to a plurality of sponsored portals located at fitness centers and non-sponsored portals the method comprising:

receiving a request from one of the sponsored or non-sponsored portals by the online wellness-related site;

processing the request at a controller to determine whether the request was received from an authorized user;

providing services to the user through the distributed network;

determining whether the request was received from one of the sponsored portals located at fitness centers; and

controlling the services available to the user based at least in part on the results of the determining step, wherein a different level of services are provided to the user based at least in

part on the results of determining if the request came from one of the sponsored portals located in a fitness center.

95. (Amended) A method as in Claim 83, further comprising the step of creating practical guidelines and advice for the control group, and wherein the services providing step comprises providing a user improvement plan for the user, the user improvement plan is selected to be similar to the practical guidelines and advice for the control group.

96. A method as in Claim 95, wherein each control group includes group result data, the method further comprising the step of:

providing result data to the portal;

storing the result data to the group result data for the authorized user's control group; and

adjusting the user improvement plan for each user in the authorized user's control group based on the stored group result data.

97. A method as in Claim 95, further comprising:

storing result data for the authorized user; and

assigning the user to a new control group based on the stored result data for the user.

: **APPENDIX B**

(Claim Charts for Independent Claims Involved in the Appeal)

81. A method of providing wellness-related services, including at least one of wellness, health, or fitness services through a publicly accessible distributed network to authorized users using authorized portals, comprising:	
providing an online site that enables wellness-related databases to be accessed from at least one of a sponsored and a non-sponsored portal;	
placing in communication at least one of a sponsored and non-sponsored portal to the online site through the publicly accessible distributed network wherein the publicly accessible distributed network includes the Internet, wherein the sponsored portal is at least in part sponsored by and located at, a	The Examiner admits that Baker et al. "do not disclose using an online system for wellness services at a fitness center, wherein one of the multiple computers is a computer residing at the fitness center and is thus sponsored at the fitness center." See Final Office Action, page 3, paragraph 4.

fitness center, and wherein at least one of the non-sponsored portals accesses the on-line site through the Internet;	3, paragraph 4.
receiving a request at the online site requesting access to the wellness-related databases;	
processing the request at the online site to determine whether the portal was sponsored and whether the request was received from an authorized user; and	
responding to the request based in part on whether the portal was sponsored and whether the user is authorized.	

85. A method of providing wellness-related services, including at least one of wellness, health, or fitness services through a publicly accessible distributed network to authorized users using authorized portals, comprising:	
providing an online site that enables wellness-related databases to be accessed from at least one of a sponsored and a non-sponsored portal; and	
placing in communication at least one of a sponsored and non-sponsored portal to the online site through a publicly available distributed network;	
wherein at least one of the sponsored portals includes a computer display located in a fitness center, wherein the method further	The examiner admits that Baker "fails to disclose obtaining data from the user through one of the portals, then providing the user with

comprises providing access to the authorized users at both the fitness center sponsored portal and the non-sponsored portals,	access to the data through the other terminal. See Final Office Action, page 4, last paragraph Through page 5, first paragraph
wherein the authorized users are able to enter fitness-related data selected from the group consisting of workout plans, workout goals, weight training plans, weight training weights and weight training repetitions at the fitness center and view the fitness data from the non-sponsored portals.	The Examiner admits that Baker et al. "do not disclose using an online system for wellness services at a fitness center, wherein one of the multiple computers is a computer residing at the fitness center and is thus sponsored at the fitness center." See Final Office Action, page 3, paragraph 4. The examiner admits that Baker "fails to disclose obtaining data from the user through one of the portals, then providing the user with access to the data through the other terminal. See Final Office Action, page 4, last paragraph Through page 5, first paragraph

86. A method of providing wellness-related services, including at least one of wellness, health or fitness services to an authorized user through a distributed communications network, comprising:	
identifying a portal with a portal identifier;	
storing the portal identifier associated with the portal in a database;	
receiving a request from the portal by an online wellness-related site; processing the request at a controller to determine whether the request was from the	

portal;	
assigning an access code to the user, the access code defining a level of wellness-related services available to the user;	
providing services to the user through the distributed network that corresponds to the user's access code;	
providing at least one control group, wherein each control group includes at least one authorized user; and	
assigning the user to one of the control groups, wherein the assigning is done automatically based on user attributes.	

90. A method as in Claim 86, wherein each control group includes group result data, the method further comprising the steps of:	
providing the result data to the portal;	
storing the result data to the group result data for the authorized user's control group; and	
adjusting the user improvement plan for each user in the authorized user's control group based on the stored group result data.	

93. A method of providing wellness-related services, including at least one of wellness, health, or fitness services through a distributed communications network, wherein the network is coupled to an on-line wellness related site and to a plurality of sponsored portals located at fitness centers and non-sponsored portals the method comprising:	The Examiner admits that Baker et al. do "not disclose using an online system for wellness services at a fitness center, wherein one of the multiple computers is a computer residing at the fitness center and is thus sponsored at the fitness center." See Final Office Action, page 3, paragraph 4.
receiving a request from one of the sponsored or non-sponsored portals by the online wellness-related site;	
processing the request at a controller to determine whether the request was received from an authorized user;	

providing services to the user through the distributed network;	
determining whether the request was received from one of the sponsored portals located at fitness centers; and	The Examiner admits that Baker et al. "do not disclose using an online system for wellness services at a fitness center, wherein one of the multiple computers is a computer residing at the fitness center and is thus sponsored at the fitness center." See Final Office Action, page 3, paragraph 4.
controlling the services available to the user based at least in part on the results of the determining step, wherein a different level of services are provided to the user based at least in part on the results of determining if the request came from one of the sponsored portals located in a fitness center.	The Examiner admits that "Baker does not disclose that the sponsored portal is located in a health or fitness." See Final Office Action, page 4, paragraph 3. Baker et al. do not teach providing a different level of services based at least in part on the results of determining if the request came from one of the sponsored portals located in a fitness center.

APPENDIX C

FINAL OFFICE ACTION

MAILED OCTOBER 8, 2002

8c 27/2376 14884.12.1

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/449,237	11/24/1999	JAMES PRESCOTT CURRY	23091/9001	6035

22859 7590 10/08/2002

FREDRIKSON & BYRON, P.A.
4000 PILLSBURY CENTER
200 SOUTH SIXTH STREET
MINNEAPOLIS, MN 55402

EXAMINER
EDELMAN, BRADLEY E

ART UNIT	PAPER NUMBER
2153	

DATE MAILED: 10/08/2002

Please find below and/or attached an Office communication concerning this application or proceeding.

Response due: 1/8/03 Final Rejection/Notice of Appeal

Docketed 10/15/02 KM

Office Action Summary	Application No.	Applicant(s)	
	09/449,237	CURRY, JAMES PRESCOTT	
	Examiner	Art Unit	
	Bradley Edelman	2153	

-- *The MAILING DATE of this communication appears on the cover sheet with the correspondence address* --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *29 July 2002* .

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *81-93 and 95-97* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *81-93 and 95-97* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☒ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

This action is in response to Applicant's request for reconsideration and amendment filed on July 29, 2002. Claims 81-93 and 95-97 are presented for further examination. Claim 94 has been cancelled.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

1. Claims 81 and 93 are rejected under 35 U.S.C. 103(a) as being unpatentable over Baker et al. (U.S. Patent No. 5,678,041, hereinafter "Baker").

In considering claim 81, Baker discloses a method of providing services through a publicly accessible distributed network (100) to authorized users using authorized portals (107-109), comprising:

providing an online site that enables databases to be accessed from at least one of multiple portals (col. 4, lines 18-35);

placing at least one of the multiple portals to the online site in communication with the on-line site through the publicly accessible distributed network (col. 4, lines 18-35), wherein the network includes the internet, and wherein at least one of the computers can access the site through the Internet (col. 3, lines 12-15);

receiving a request at the online site requesting access to the databases (col. 4, lines 17-19);

processing the request to determine which portal the request came from (col. 4, lines 9-12, 17-19, "ID107, ID108, ID109," "identity of the requesting user terminal"), and whether the request was received from an authorized user (col. 4, lines 24-30, "user clearances 107-109"); and

responding to the request based on which portal the request came from and whether the request was received from an authorized user (col. 4, lines 17-30).

However, Baker does not expressly disclose using the online system for wellness services at a fitness center, wherein one of the multiple computers is a computer residing at the fitness center and is thus sponsored by the fitness center. Nonetheless, the claim limitations that the database is a wellness-related database, and that one of the computers may be at a fitness center, and thus would be sponsored by the fitness center, are only found in the nonfunctional descriptive material and are not functionally involved in the steps recited. It would make no difference if one of the computers were at a school or business (as disclosed by Baker – col. 1, lines 46-55), a health center, a neighbor's house, a library, or any other location, or if the data was wellness-related, financial-related, or adult-material-related. The terminal ID determination and the authorization steps would be performed the same regardless of the type of information being accessed or the location of the requesting computer. Thus, this descriptive material will not distinguish the claimed invention from the prior art in terms of

patentability, *see In re Gulack*, 703 F.2d 1381, 1385, 217 USPQ 401, 404 (Fed. Cir. 1983); *In re Lowry*, 32 F.3d 1579, 32 USPQ2d 1031 (Fed. Cir. 1994).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to include any type of information and to place one of the terminals at any location in the system taught by Baker, because access to any type of personal or explicit information should be restricted to prevent tampering or unauthorized access. The subjective type of information and location does not functionally relate to the steps in the method claimed, and thus does not patentably distinguish the claimed invention.

Claim 93 contains no further limitations over claim 81, except that a different level of services are provided to the user based at least in part on the results of determining which portal the request came from. Nonetheless, this feature is taught in col. 4, lines 9-30 of Baker ("unlimited access, restricted use . . ."). Although Baker does not disclose that the sponsored portal is located in a health or fitness center, that claim limitation does not functionally relate to the steps in the method claimed, and thus does not patentably distinguish the claimed invention, as discussed above.

2. Claims 82-92 and 95-97 are rejected under 35 U.S.C. 103(a) as being unpatentable over Baker, in view of Szabo (U.S. Patent No. 5,954,640).

In considering claim 82, although the system taught by Baker discloses substantial features of the claimed invention, it fails to disclose obtaining data from the

user through one of the portals, and then providing the user with access to the data through the other terminal. However, these features are well known for online sites, and particularly for wellness-related sites, as evidenced by Szabo. In a similar art, Szabo discloses a system for accessing an online wellness-related site, wherein users can enter information into the site to be stored at a database, and wherein the user can later retrieve that data from one of many computers (including a sponsored kiosk). See col. 6, lines 5-9; col. 3, lines 56-61. Thus, given the teaching of Szabo, a person having ordinary skill in the art would have readily recognized the desirability and advantages of allowing users to input data, as taught by Szabo, into the information access system taught by Baker, to allow greater interactivity among users, thus providing greater opportunity for market growth of the online site. Therefore, it would have been obvious to allow inputting data into the system, as taught by Szabo, in the system taught by Baker. In addition, both Szabo and Baker teach that multiple computers can be used to retrieve data (see Baker, Fig. 1) and/or to input data (see Szabo, col. 4, lines 16-23).

In considering claim 83, Szabo further discloses automatically assigning the user to a control group based on user attributes (col. 9, line 66 – col. 10, line 9; "the models themselves may be adaptive based on the experiences of individual users or groups . . . neural network technology and other adaptive paradigms may be employed to dynamically improve the models through use and feedback.").

In considering claim 84, Szabo further discloses providing fitness advice and goals to the group, wherein the advice and goals are at least in part a result of the group result data (col. 10, lines 1-34).

Claim 85 contains no further functional limitations over claims 81 and 82 combined and is rejected for the same reasons as stated above. Although the amendment to claim 85 has added that authorized users are able to enter fitness-related data "selected from the group consisting of workout plans, workout goals, weight training plans, weight training weights and weight training repetitions," this information is again non-functional, descriptive material that does not add to the functional operation of the claimed invention. The entered data could have been any data (i.e. business plan data, school-related program data, etc.) and the invention would still perform the same function – that is, allowing data to be entered at one portal in the system, and allowing retrieval and viewing of the data at a different portal.

In considering claim 86, Baker discloses a method of providing services to an authorized user through a distributed communications network, comprising:

identifying a portal with a portal identifier (ID107, ID108, ID109), and storing the portal identifier in a database (col. 4, lines 20-22);

receiving a request from the portal by an online site, and processing the request at a controller to determine whether the request was from the portal (col. 4, lines 1-16); and

assigning an access code to the user (clearance 107, clearance 108, clearance 109), the access code defining a level of services available to the user, and then providing services to the user through the distributed network that correspond to the user's access code (col. 4, lines 25-32).

However, Baker fails to disclose the use of the access system for wellness-related information and services, and Baker further fails to disclose providing at least one control group, wherein each control group includes at least one authorized user, and assigning the user to one of the control groups, wherein the assigning is done automatically based on user attributes. Nonetheless, the use of user authorization routines for wellness related sites is well known, as evidenced by Szabo. In a similar art, Szabo discloses a wellness-related site that provides wellness-related access and services to users, wherein the users must identify themselves before gaining access to the data (col. 13, lines 43-47, 55-57, "safeguards are also placed to prevent unauthorized intrusion into an individual's personal information records"). Thus, it would have been obvious to a person having ordinary skill in the art to use the safeguards taught by Baker for a wellness-related site, such as taught by Szabo, because users would not want others to know their personal medical and health information.

Szabo further discloses providing at least one control group, wherein each control group includes at least one authorized user, and assigning the user to one of the control groups, wherein the assigning is done automatically based on user attributes (col. 9, line 66 – col. 10, line 9; wherein a neural network may be employed to dynamically update the group information through the use of a feedback loop).

In considering claim 87, Szabo further discloses providing information or goods to the user based upon the control group to which the user has been assigned (col. 9, line 66 – col. 10, line 9).

In considering claims 90 and 96, Szabo further discloses that each control group includes group result data, the method further comprising providing the result data to the portal, storing the result data to the group result data for the authorized user's control group, and adjusting the user improvement plan for each user in the authorized user's control group based on the stored group result data (col. 10, lines 1-34).

In considering claim 95, Szabo further discloses creating practical guidelines and advice for the control group, including a user improvement plan selected to be related to the guidelines and advice (col. 10, lines 1-34).

In considering claim 97, Szabo further discloses storing result data for the authorized user, and assigning the user to a new control group based on the stored result data for the user (col. 10, lines 14-34).

3. Claims 88, 89, 91, and 92 are rejected under 35 U.S.C. 103(a) as being unpatentable over Baker, in view of Szabo, and further in view of Roth (U.S. Patent No. 5,890,997).

In considering claim 88, Szabo further discloses creating practical guidelines and advice for the control group, including a user improvement plan selected to be related to the guidelines and advice (col. 10, lines 1-34).

However, Szabo does not disclose that the guidelines are workout guidelines and the advice is workout advice. Nonetheless, the use of workout data to create practical guidelines and advice for groups of users is well known, as evidenced by Roth. In a similar art, Roth teaches a computerized fitness management system for multiple users at a health club which includes a database storing wellness-related information for each user (cols. 28-30), wherein workout data is used to create practical guidelines and advice for groups of users in the system, and wherein such data can be used to update workout routines and other wellness-related advice in the system (col. 3, lines 47-52, 59-63; col. 4, lines 3-10). Thus, given the teaching of Roth, a person having ordinary skill in the art would have readily recognized the desirability and advantages of allowing updating of the data in the combined teaching of Baker and Szabo based on workout data and results, as taught by Roth, so that medication and other advice can be altered in response to the current state of a user's health. Therefore, it would have been obvious to change the guidelines and advice in the system taught by Baker and Szabo according to workout results, as taught by Roth.

In considering claim 89, Szabo further discloses that the improvement plan is at least in part based on the collective attributes of the control group (col. 10, lines 1-34). Again, in view of the workout updating features taught by Roth, as discussed above, it

would have been obvious to a person having ordinary skill in the art to base a user improvement plan on collective group workout data, so that group advice and information can be based on the current state of the collective users' health.

In considering claim 91, Szabo further discloses checking if the user improvement plan for users in the control group needs to be adjusted (col. 12, lines 45-52). Thus, although an "alarm" is not expressly disclosed, Examiner takes official notice that alarms are notoriously well known in the computer art as a means for reminding or warning users of particular events. Therefore, it would have been obvious to a person having ordinary skill in the art to provide an alarm signal to the system administrator if adjustments are needed, to make a human aware of potentially harmful drug interactions or other health risks.

In considering claim 92, Szabo further discloses storing result data for the authorized user, and assigning the user to a new control group based on the stored result data for the user (col. 10, lines 14-34).

Response to Arguments

In response to applicant's request for reconsideration filed on July 29, 2002, the following arguments are noted:

a. Examiner has given the sponsorship of the sponsored portal no weight, and Examiner has read all of the business method limitations out of claims 81 and other claims of the presently recited invention.

b. Baker does not disclose methods for providing wellness-related services from an online site to both sponsored portals sponsored by and located at a fitness center and nonsponsored portals, where responding to information requests depends on whether the portal was sponsored.

c. Neither Baker nor Szabo disclose a method for providing wellness-related services including allowing authorized users to enter fitness-related data selected from the group consisting of workout plans, workout goals, weight training plans, weight training weights and weight training repetitions at a fitness center and to view the fitness data from the nonsponsored portals, as recited in claim 85.

d. Szabo does not disclose assigning users to groups automatically based on user attributes, as claimed in claim 86.

In considering (a), Applicant contends that Examiner has given the sponsorship of the sponsored portal no weight, and Examiner has read all of the business method limitations out of claims 81 and other claims of the presently recited invention. Examiner respectfully disagrees.

Examiner has not ignored either the sponsorship or the business method aspects of the invention. The Baker reference relied on is intended for use in a system including sponsored and non-sponsored computers, as discussed in the background section of

the patent. The reference describes two separate systems that include sponsored and non-sponsored portals: 1) A school that sponsors portals allowing access to certain resources, and 2) Businesses that would like to allow their employees to access only work-related resources (col. 1, lines 46-55). Thus, the terminals described in column 4 of the Baker system that allow different levels of access to databases would either be part of a sponsored system, such as a school or business, or would be non-sponsored, and thus not part of the school or business system. Thus, both the sponsorship aspects and the business method aspects of Applicants claimed invention are disclosed in the Baker reference.

In considering (b), Applicant contends that Baker does not disclose methods for providing wellness-related services from an online site to both sponsored portals sponsored by and located at a fitness center and nonsponsored portals, where responding to information requests depends on whether the portal was sponsored. Examiner agrees that Baker does not disclose that the sponsored portals are located at a fitness center and that the services are wellness-related. However, as stated in the rejections above, these limitations are non-functional and are merely descriptive, and thus do not distinguish the claimed invention from the prior art in terms of patentability. Regarding functionality of the claimed invention, it makes no difference if the portals are placed in a health center, school, or business, and it makes no difference whether the data is wellness-related, educational, or business-oriented. The invention will still

perform the same steps and functions, all of which are disclosed by Baker described in the rejections above.

In considering (c), Applicant contends that neither Baker nor Szabo disclose a method for providing wellness-related services including allowing authorized users to enter fitness-related data selected from the group consisting of workout plans, workout goals, weight training plans, weight training weights and weight training repetitions at a fitness center and to view the fitness data from the nonsponsored portals, as recited in claim 85. Examiner agrees. However, again, the descriptive, non-functional nature of the data included in the claimed system does not render the system patentable, for the reasons stated in (b) above.

In considering (d), Applicant contends that Szabo does not disclose assigning users to groups automatically based on user attributes, as claimed in amended claim 86. Examiner respectfully disagrees. Column 10, lines 1-9 state, "the models themselves may be adaptive based on the experiences of individual users or groups . . . neural network technology and other adaptive paradigms may be employed to dynamically improve the models through use and feedback." Thus, Szabo discloses automatically basing models based user attributes, and applying these models when forming groups.

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Bradley Edelman whose telephone number is (703) 306-3041. The examiner can normally be reached on Monday to Friday from 8:30 AM to 5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Glen Burgess can be reached on (703) 305-4792. The fax phone numbers for the organization where this application or proceeding is assigned are as follows:

For all After Final papers: (703) 746-7238.

For all other correspondences: (703) 746-7239.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 308-3900.

GLENTON B. BURGESS
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

BE
October 3, 2002

Notice of References Cited

Application/Control No.	Applicant(s)/Patent Under Reexamination	
09/449,237	CURRY, JAMES PRESCOTT	
Examiner	Art Unit	Page 1 of 1
Bradley Edelman	2153	

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-5,890,997	04-1999	Roth, Eric S.	482/8
	B	US-6,053,844	04-2000	Clem, William	482/1
	C	US-5,916,063	06-1999	Alessandri, Nerio	482/1
	D	US-5,911,687	06-1999	Sato et al.	600/300
	E	US-6,112,224	08-2000	Peifer et al.	340/3.1
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S			-		
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	PR Newswire, Concentric Network Powers Netpulse Fitness Machines in Clubs Across the Country, PR Newswire, New York, September 9, 1998, p. 1.
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

APPENDIX D

(U.S. PATENT NO. 5,678,041, "BAKER")

US005678041A

United States Patent [19]

Baker et al.

[11] Patent Number: 5,678,041

[45] Date of Patent: Oct. 14, 1997

[54] **SYSTEM AND METHOD FOR RESTRICTING USER ACCESS RIGHTS ON THE INTERNET BASED ON RATING INFORMATION STORED IN A RELATIONAL DATABASE**

[75] Inventors: **Brenda Sue Baker**; **Eric Grosse**, both of Berkeley Heights, N.J.

[73] Assignee: **AT&T**, Middletown, N.J.

[21] Appl. No.: **519,268**

[22] Filed: **Aug. 25, 1995**

Related U.S. Application Data

[63] Continuation-in-part of Ser. No. 469,276, Jun. 6, 1995, abandoned.

[51] **Int. Cl.**[6] **G06F 17/30**

[52] **U.S. Cl.** **395/609**; 395/601; 395/610; 395/188.01

[58] **Field of Search** 395/600, 726, 395/739, 188.01, 187.01, 609, 610, 601

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,961,224	10/1990	Yung	380/25
5,113,499	5/1992	Ankney et al.	340/825.34

OTHER PUBLICATIONS

Jay Friedland, "Surfwatch Blocks Internet Pronography, Breaktrough Product Ships Today"; obtained at the internet address http://www.surf.com May 15, 1995.

Albert Lunde, "Parental Control: A Proposed Addition to HTTP"; obtained through internet newsgroup article 1060 of comp.infosystems.www.servers.unix May 29, 1995.

The Little Schoolhouse Logs On, PC Magazine, Feb. 21, 1995, p. 62.

The Rating Game, Stanford Observer, Winter 1995, p. 13.

Surf-Watch World Wide Web Homepage, copyright 1995 SurfWatch Software, Inc.; obtained at the Internet address http://www.surfcom/index.html on Jun. 6, 1995.

Re: Parental Control: A Proposed Addition to HTTP, Albert Lunde, Article 1060 of comp.infosystems.www.servers.unix newsgroup on the World Wide Web, May 29, 1995.

Primary Examiner—Thomas G. Black
Assistant Examiner—Greta L. Robinson
Attorney, Agent, or Firm—Michele Conover

[57] **ABSTRACT**

A system and method for selectively controlling database access by providing a system and method that allows a network administrator or manager to restrict specific system users from accessing information from certain public or otherwise uncontrolled databases (i.e., the WWW and the Internet). The invention employs a relational database to determine access rights, and this database may be readily updated and modified by an administrator. Within this relational database specific resource identifiers (i.e., URLs) are classified as being in a particular access group. The relational database is arranged so that for each user of the system a request for a particular resource will only be passed on from the local network to a server providing a link to the public/uncontrolled database if the resource identifier is in an access group for which the user has been assigned specific permissions by an administrator. In one preferred embodiment, the invention is implemented as part of a proxy server within the user's local network.

23 Claims, 6 Drawing Sheets

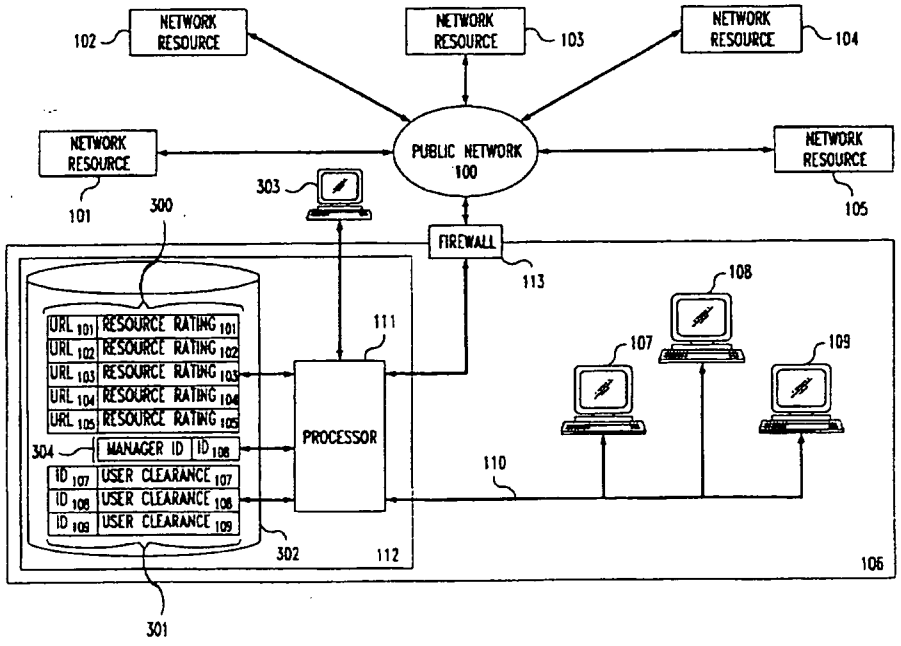


FIG. 1

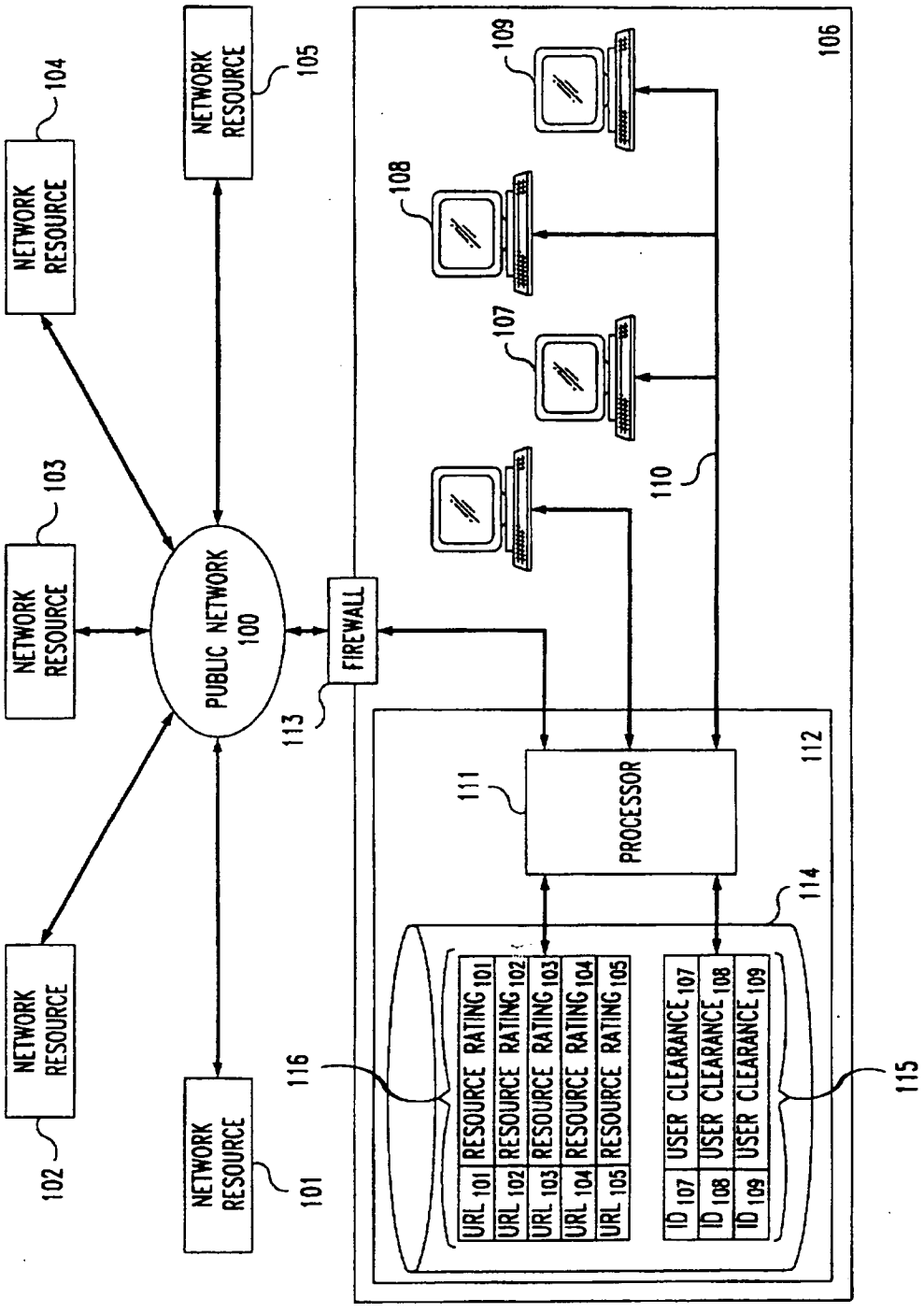

NETWORK RESOURCE
NETWORK RESOURCE
NETWORK RESOURCE
NETWORK RESOURCE
NETWORK RESOURCE
PUBLIC NETWORK 100
FIREWALL
PROCESSOR
URL 101 RESOURCE RATING 101
URL 102 RESOURCE RATING 102
URL 103 RESOURCE RATING 103
URL 104 RESOURCE RATING 104
URL 105 RESOURCE RATING 105
ID 107 USER CLEARANCE 107
ID 108 USER CLEARANCE 108
ID 109 USER CLEARANCE 109
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116

FIG. 2

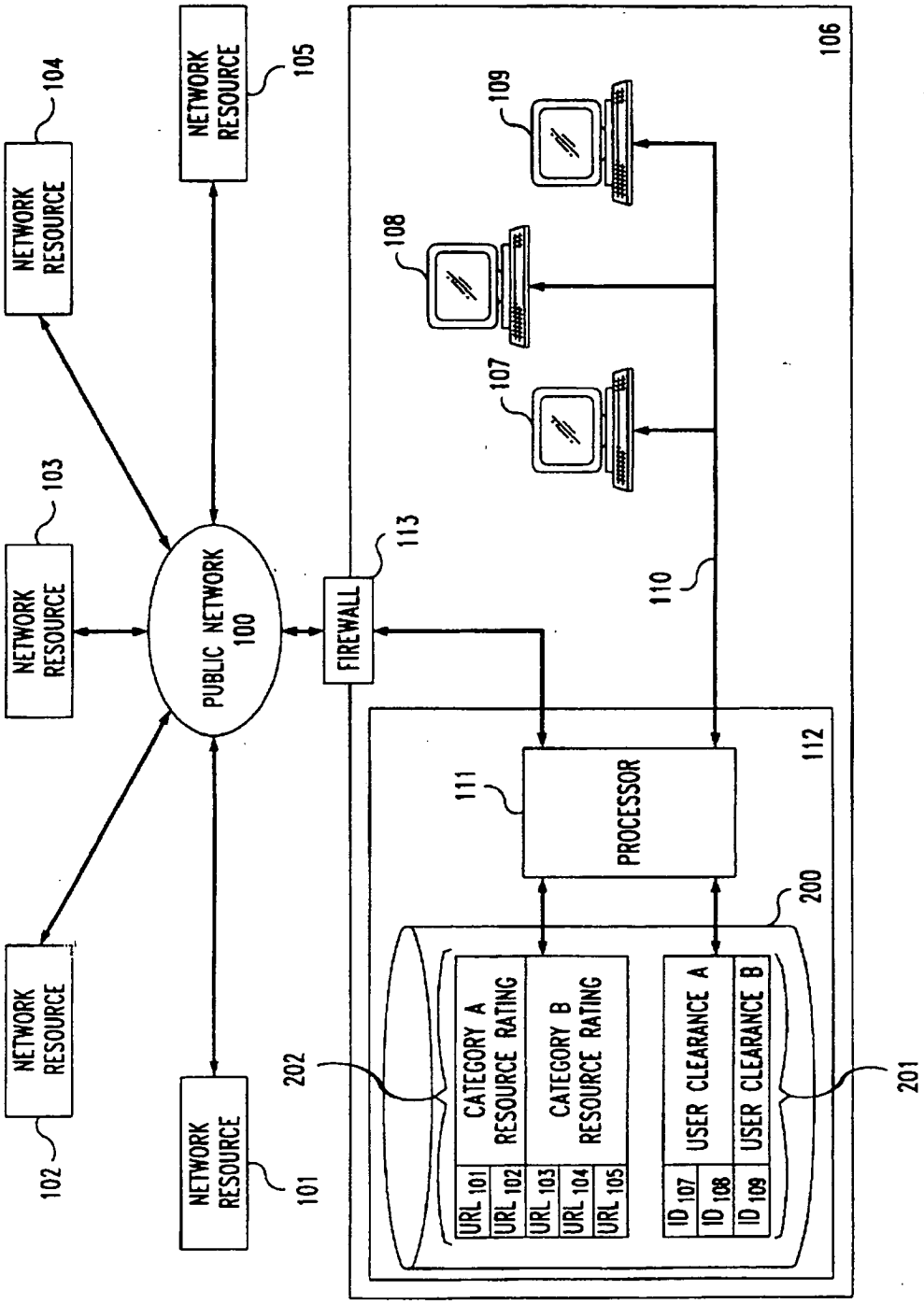

NETWORK RESOURCE
NETWORK RESOURCE
NETWORK RESOURCE
NETWORK RESOURCE
NETWORK RESOURCE
PUBLIC NETWORK 100
FIREWALL
PROCESSOR
CATEGORY A RESOURCE RATING
CATEGORY B RESOURCE RATING
URL 101
URL 102
URL 103
URL 104
URL 105
USER CLEARANCE A
USER CLEARANCE B
ID 107
ID 108
ID 109
101
102
103
104
105
106
107
108
109
110
111
112
113
200
201
202

FIG. 3

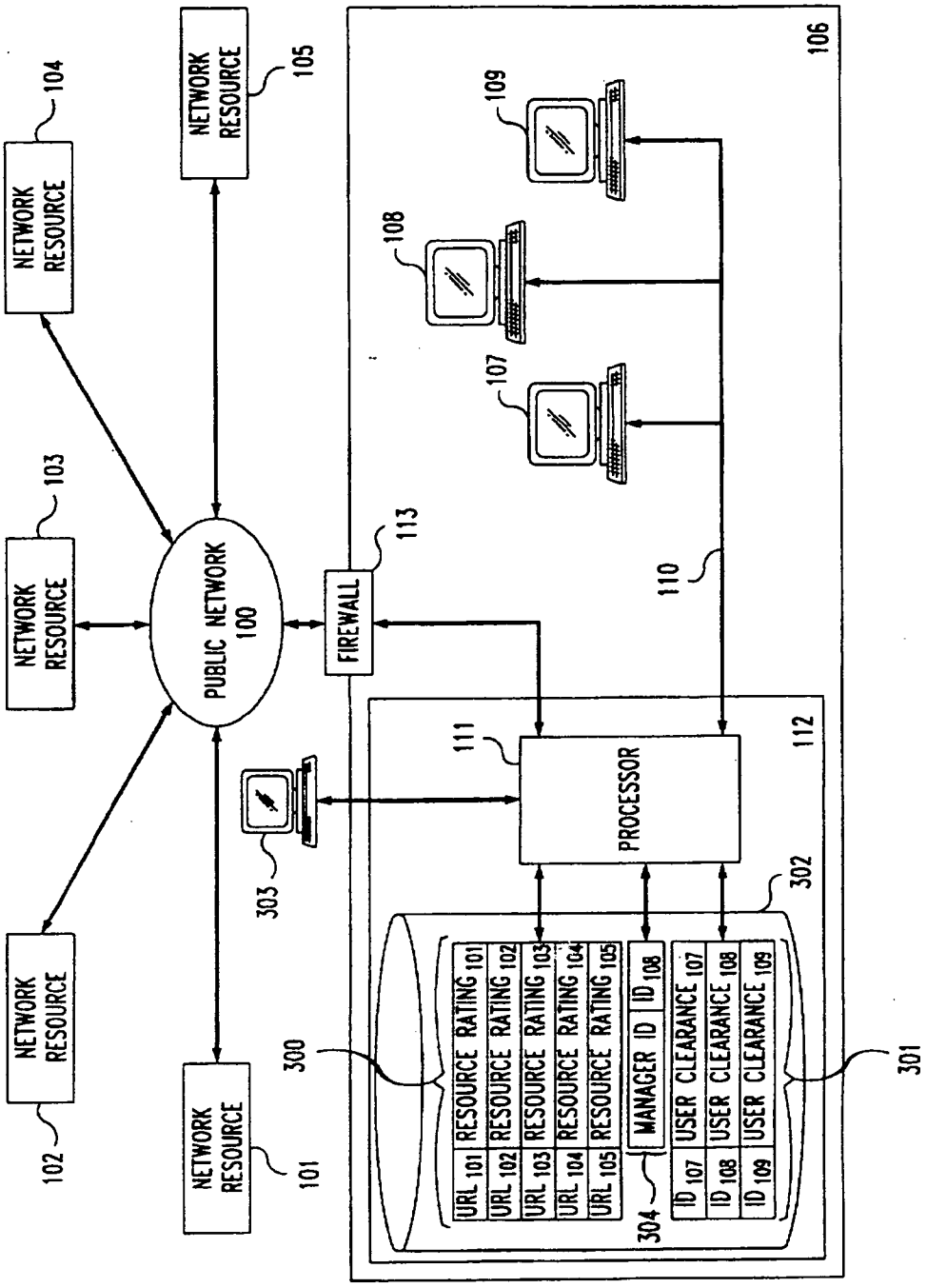

NETWORK RESOURCE
102
NETWORK RESOURCE
103
NETWORK RESOURCE
104
NETWORK RESOURCE
101
NETWORK RESOURCE
105
PUBLIC NETWORK 100
FIREWALL
113
106
109
108
107
110
111
PROCESSOR
112
303
302
300
301
URL 101
RESOURCE RATING 101
URL 102
RESOURCE RATING 102
URL 103
RESOURCE RATING 103
URL 104
RESOURCE RATING 104
URL 105
RESOURCE RATING 105
MANAGER ID
ID 108
304
ID 107
USER CLEARANCE 107
ID 108
USER CLEARANCE 108
ID 109
USER CLEARANCE 109

FIG. 4

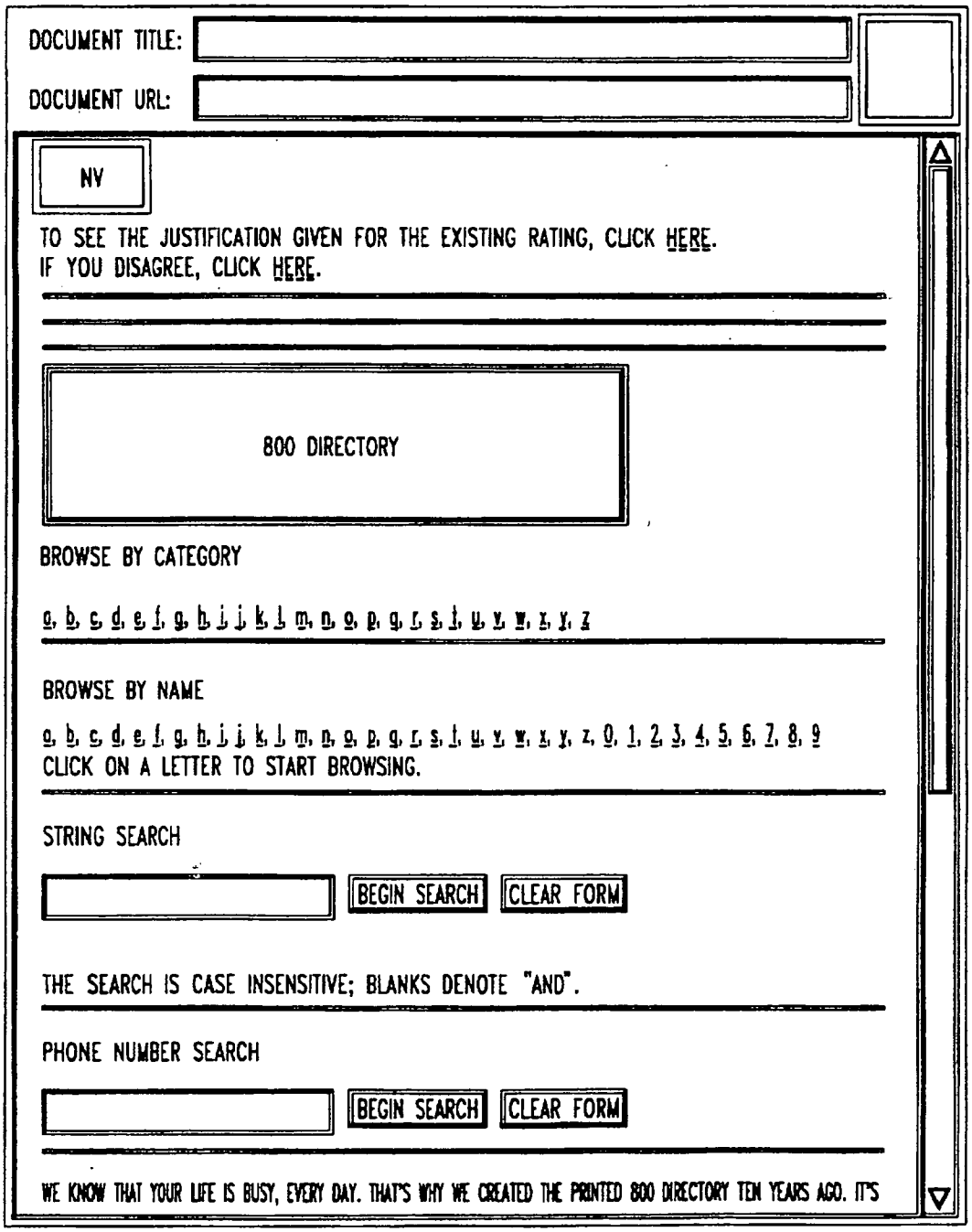

DOCUMENT TITLE:
DOCUMENT URL:
NV
TO SEE THE JUSTIFICATION GIVEN FOR THE EXISTING RATING, CLICK HERE.
IF YOU DISAGREE, CLICK HERE.
800 DIRECTORY
BROWSE BY CATEGORY
a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z
BROWSE BY NAME
a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
CLICK ON A LETTER TO START BROWSING.
STRING SEARCH
BEGIN SEARCH
CLEAR FORM
THE SEARCH IS CASE INSENSITIVE; BLANKS DENOTE "AND".
PHONE NUMBER SEARCH
BEGIN SEARCH
CLEAR FORM
WE KNOW THAT YOUR LIFE IS BUSY, EVERY DAY. THAT'S WHY WE CREATED THE PRINTED 800 DIRECTORY TEN YEARS AGO. IT'S

FIG. 5

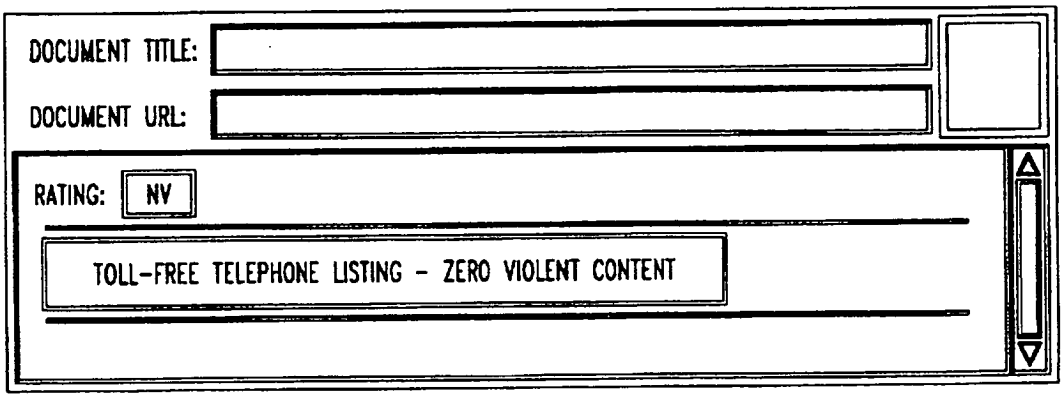
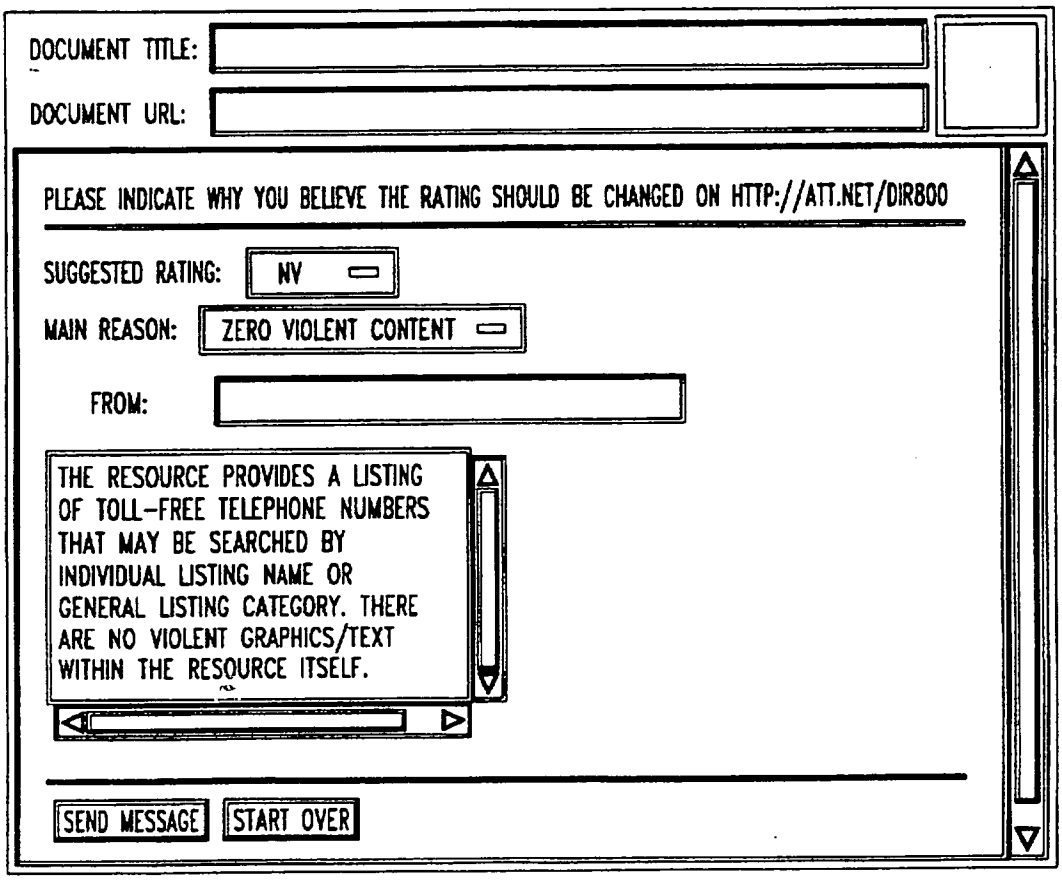

DOCUMENT TITLE:
DOCUMENT URL:
RATING:
NV
TOLL-FREE TELEPHONE LISTING - ZERO VIOLENT CONTENT

FIG. 6


DOCUMENT TITLE:
DOCUMENT URL:
PLEASE INDICATE WHY YOU BELIEVE THE RATING SHOULD BE CHANGED ON HTTP://ATT.NET/DIR800
SUGGESTED RATING:
NV
MAIN REASON:
ZERO VIOLENT CONTENT
FROM:
THE RESOURCE PROVIDES A LISTING OF TOLL-FREE TELEPHONE NUMBERS THAT MAY BE SEARCHED BY INDIVIDUAL LISTING NAME OR GENERAL LISTING CATEGORY. THERE ARE NO VIOLENT GRAPHICS/TEXT WITHIN THE RESOURCE ITSELF.
SEND MESSAGE
START OVER

SYSTEM AND METHOD FOR RESTRICTING USER ACCESS RIGHTS ON THE INTERNET BASED ON RATING INFORMATION STORED IN A RELATIONAL DATABASE

CROSS-REFERENCE TO RELATED APPLICATION

This is a continuation-in-part of the U.S. patent application Ser. No. 08/469,276, filed on Jun. 6, 1995 entitled "System And Method For Database Access Administration", now abandoned.

TECHNICAL FIELD

The invention relates to controlling database access and, more particularly, to selectively providing such control with respect to otherwise public databases.

BACKGROUND OF THE INVENTION

Files or other resources on computers around the world may be made publicly available to users of other computers through the collection of networks known as the Internet. The collection of all such publicly available resources, linked together using files written in Hypertext Mark-up Language ("HTML"), is known as the World Wide Web ("WWW").

A user of a computer that is connected to the Internet may cause a program known as a client to request resources that are part of the WWW. Server programs then process the requests to return the specified resources (assuming they are currently available). A standard naming convention has been adopted, known as a Uniform Resource Locator ("URL"). This convention encompasses several types of location names, presently including subclasses such as Hypertext Transport Protocol ("http"), File Transport Protocol ("ftp"), gopher, and Wide Area Information Service ("WAIS"). When a resource is downloaded, it may include the URLs of additional resources. Thus, the user of the client can easily learn of the existence of new resources that he or she had not specifically requested.

The various resources accessible via the WWW are created and maintained by many different people on computers around the world, with no centralized control of content. As particular types of information or images contained in this uncontrolled information collection may not be suitable for certain users, it may be desirable to selectively restrict access to WWW resources. For example, parents or school teachers might wish to have children access useful information, but not obscene material (which the children may be exposed to as a result of innocent exploration of the WWW, or through the incidental downloading of a URL). Another example is the case of school teachers who would like their students to access just a particular group of resources during a class meeting. A third example is businesses that would like their employees to access only work-related resources, but not to spend their time on other WWW explorations. In general, a particular user might need to be restricted to different resources at different times, as in the case of a student restricted to different sets of resources during classes on different subjects.

Some authorities such as schools ask the users to abide by a policy statement by which they agree to restrict their exploration of the WWW, for example, by agreeing not to download obscene material. However, voluntary compliance with such a policy will not prevent the accidental downloading of resources that are not readily identifiable as forbidden or inappropriate prior to downloading and viewing.

Naturally, technical solutions such as "firewalls" are also available to limit or impede access to the WWW and Internet. These firewalls are software-based gateways that are commonly installed to protect computers on a local area network ("LAN") from being attacked by outsiders. One effect of installing a firewall is that WWW clients can no longer directly contact WWW servers. Typically, this proves too restrictive, and users resort to "proxy servers" that are directly contacted by WWW clients. These proxy servers have special abilities to forward requests through the firewall, and thereby provide communication to and from servers on the Internet. For efficiency, a proxy server may also cache some resources locally. Current clients and proxy servers yield access to every public resource in the WWW—. They are not configured to allow a particular user to request some resources, while preventing access by that user to other resources.

Some "filtering" of the available WWW resources may be effected within systems that offer indirect access. In these systems an information provider would download resources from the WWW and maintain copies of the resources. Users would access these copies. The information provider can review the resources as they are obtained from the WWW, and edit out any inappropriate or obscene material prior to making the resource available to users. A disadvantage of this scheme is that the material provided by the information provider may be out-of-date compared to the original resource on the WWW.

In an alternate scheme of "filtered" access to WWW resources, a proxy server provides a user with a menu of allowed resources that may be accessed, and users can obtain any resources that can be reached by a series of links from the menu resources. The user is only permitted to request URLs via this menu. This particular method has two disadvantages. First, many resources must be excluded from the menu because they contain links to inappropriate material, even though they themselves might be acceptable. Second, a resource may change over time to include new links that might lead to inappropriate material, and thereby provide a user with an unintended pathway of access to such.

In still another method of "filtered" access to WWW resources, the client or proxy server checks each resource for a list of disallowed words (i.e.; obscenities; sexual terms, etc.) and shows the user only those resources that are free of these words. However, this method does not permit filtering of images and does not prohibit resources that might be inappropriate due to content other than specific words.

Yet another means of protecting users from inappropriate or obscene materials has been established by the computer and video game manufacturers. The games are voluntarily rated on the dimensions of violence, nudity/sex, and language. Although such conventions have not yet been adopted in the WWW, the analog would be to add such ratings to WWW resources, presumably with digital signatures to prevent forgery. A WWW client could then, if so programmed, choose not to save or display any resource that is unrated or has an unacceptable rating for the given audience. The disadvantage of this scheme is the need to convince the many people who provide useful servers (often on a non-professional or pro bono basis) to coordinate with a rating panel.

All of the present systems for limiting user access to an uncontrolled public database resources, such as those available on the WWW, have obvious shortcomings. Presently, there exists no simple means for an authority (i.e.; teacher, supervisor, system administrator, etc.) to selectively control

WWW access by one or more users, without significantly impairing the users' ability to communicate with the Internet. This is especially true if the particular authority wishing to exert such control has few computer skills with respect to the management of information/services networks.

SUMMARY OF THE INVENTION

The present invention overcomes the deficiencies of prior schemes for regulating network database access by providing a system and method that allows one or more network administrators/managers to rate particular information and/or services. This rating is then employed to restrict specific system users from accessing the information/services via certain public or otherwise uncontrolled databases (i.e., the WWW and the Internet). The invention employs a relational database to determine access rights, and store rating information. The rating information database may be readily updated and modified by an administrator/manager. Within this relational database specific resource identifiers (i.e., URLs) are classified as being associated with a particular access rating. The relational database is arranged so that for each user of the system a request for a particular resource will only be passed on from the local network to a server providing a link to the public/uncontrolled database if the resource identifier has an access rating for which the user has been assigned specific permissions by an administrator/manager. In one preferred embodiment, the invention is implemented as part of a proxy server within the user's local network. In another embodiment, the system maintains a ratings resource file associated with each specific resource identifier, wherein comments, conditions, etc. relating the particular resource are stored.

BRIEF DESCRIPTION OF THE DRAWING

In the drawing:

FIG. 1 is a simplified diagram of an exemplary system embodying the invention;

FIG. 2 is a simplified diagram of an alternate arrangement of the system of FIG. 1 adapted to facilitate the classification of URLs into rating groups;

FIG. 3 is a simplified diagram of an alternate arrangement of the system of FIG. 1 including system management adaptations;

FIG. 4 is an illustration of ratings information returned to a system manager upon retrieval of a particular network resource;

FIG. 5 is an illustration of resource categorization information provided to a network manager; and

FIG. 6 is an illustration of a ratings editing page accessible by a network manager.

DETAILED DESCRIPTION OF THE INVENTION

FIG. 1 is a simplified diagram of an exemplary system embodying the invention. A related system is the subject of the co-pending, and commonly assigned U.S. patent application Ser. No. 08/469,342, entitled "System And Method For Database Access Control" which was filed on Jun. 6, 1995. As shown in FIG. 1, the system includes public network **100**, network resources **101–105**, and user site **106**. Particular users at user site **106** gain access to public network **100** via user terminals **107**, **108** and **109**. Each of these user terminals is linked by local area network ("LAN") **110** to processor **111** within proxy server **112**. Finally, proxy server **112** provides a connection from processor **111** to public network **100** via firewall **113**.

Requests from user terminals **107–109** for access to network resources (**101–105**) through public network **100** are submitted to processor **111** within proxy server **112**. In this particular embodiment of the invention, the submitted requests are assumed to be in the form of URLs. As is well known in art, when URLs are submitted to a proxy server, the particular requesting user terminal is identified to the proxy server by an identification header attached to the URL. For the system shown in FIG. **1**, the identification code for user terminal **107** is ID_{107}, the identification code for user terminal **108** is ID_{108}, and the identification code for user terminal **109** is ID_{109}. In addition, within the system of FIG. **1**, URLs designated as URL_{101}, URL_{102}, URL_{103}, URL_{104} and URL_{105}, represent requests for information from network resources **101**, **102**, **103**, **104** and **105**, respectively.

Upon receipt of an incoming URL, processor **111** is programmed to determine the identity of the requesting user terminal from the URL header. This identification information is then utilized by processor **111** to cross-reference the received URL with information stored in relational database **114**. Relational database **114** contains listing **115** which associates each of the user identification codes (ID_{107}, ID_{108} and ID_{109}) with a user clearance code (user $clearances_{107}$, user $clearances_{108}$ and user $clearances_{109}$, respectively). These user clearances indicate the particular rating class or classes of network resources that a given user terminal is allowed to access (i.e.; unlimited access; restricted use of URLs identified as accessing violent subject matter; restricted use of URLs that are identified as accessing obscene subject matter; etc). Also contained in relational database **114** is listing **116** which includes a register of allowable URLs ($URL_{101-105}$) that may be transmitted from a user terminal to access network resources. Listing **116** associates each of these URLs with a particular resource rating data (resource $rating_{101-105}$). The resource rating associated with each of said URLs can be something as simple as a rating class indicator. For example, an indication that a particular URL is approved for use by all users, or that use of a particular URL is restricted for some reason (i.e.; the URL accesses network resources that contain violent or obscene subject matter).

For example, assume that a system administrator or manager had subjectively categorized the network resources of FIG. **1** into three classes (non-violent—NV, moderately violent—MV, and violent—V) as follows: network resource **101**—NV, network resource **102**—NV, network resource **103**—NV, network resource **104**—MV, and network resource **105**—V). The URL/resource rating listing **116** would then contain the following data:

URL	Resource Rating
URL_{101}	NV
URL_{102}	NV
URL_{103}	NV
URL_{104}	MV
URL_{105}	V

Further assume that user terminal **107** should be allowed access to all network resources (NV, MV and V); that user terminal **108** should only be allowed to access NV and MV rated resources; and that user terminal **109** should be allowed to access only NV resources. Information reflective of these user terminal clearances would be stored within listing in **115** as follows:

User Identification	User Clearance
ID_{107}	NV, MV, V
ID_{108}	NV, MV
ID_{109}	NV

Within the system of FIG. 1, when a requesting user terminal transmits a URL via LAN **110**, processor **111** receives the URL and the requesting user terminal identification code. Processor **111** then queries listing **115** to determine the allowable resource ratings for the particular requesting user terminal, and listing **116** to determine the resource rating of the network resource that will be accessed by the particular received URL. If a URL requesting network resource **101** was received by processor **111** from user terminal **107**, list **115** and **116** within relational database **114** would yield information indicating that user terminal **107** was cleared to access NV, MV and V rated network resources, and that URL_{101} had a rating of NV. As the rating of the requested resource was one of the ratings for which the requesting user terminal had clearance, processor **111** would forward the request for information (URL_{101}) to public network **100** via firewall **113**. Assuming the requested resource was available, public network returns the requested information to user terminal **107** via firewall **113**, processor **111** and LAN **110**. Contrastingly, if a URL having a rating that the requesting user terminal is not cleared for is received by processor **111**, that request for information is denied. For instance, if URL_{105} is received by processor **111** from user terminal **109**, relational database **114** is accessed. Since the data within listings **115** and **116** show that URL_{105} has a rating of V, and that user terminal **109** is cleared to access only NV rated network resources, processor **111** denies the request for information, and no URL is sent to public network **100**. Processor **111** could also be programmed to deny all requests from user terminals for un-rated resources. This would prohibit the accessing of network resources that had not been reviewed or rated by the system administrator/manager. It will also be understood from the above description of the invention that images contained within a given resource (i.e., in-line images) are subject to the same rating given to the resource. There would be no need to rate the in-line images separately.

In the particular embodiment described above, relational database **114** stores a list of user terminal identification codes and the various user clearances reflective of the ratings of network resources that each user terminal should be allowed to retrieve from public network **100**. It will be understood that the invention could be modified so that the list of user clearances associated with a given user terminal identification code serves as a restrictive list (i.e.; that user is not allowed to retrieve network resources having that rating). This restrictive listing functionality could be readily facilitated by reprogramming processor **111**. In addition, the invention could be modified so that the identification codes recognized by processor **111** and stored in relational database **114** are user specific, as opposed to user terminal specific. In other words, the system of FIG. **1** could be modified so that a given individual using a terminal is identified to the system by a personal password or other identifying code. Access or denial of the transmission of particular URLs is effected by the system as a function of that person's identity, regardless of the particular user terminal they may be utilizing.

The above described system may also be modified so that URLs are identified as being in a rating category within the memory structure of a relational database. FIG. **2** provides a simplified diagram of a system similar to that of FIG. **1**, but adapted to facilitate the classification of URLs into rating groups. As shown, relational database **200** includes user identification code listing **201** and URL listing **202**. Listing **201** designates user identification codes ID_{107} and ID_{108} as being in the user clearance A category, and ID_{109} as being in the user clearance B category. Upon receipt of an incoming URL, processor **111** ascertains the identity of the requesting user terminal from the URL header, and then utilizes this identification information to determine the clearance category specified for that particular user within listing **201**. The particular URL received by processor **111** is then cross-referenced with listing **202** to determine the associated resource rating category. If the requesting user has a clearance that corresponds to resource rating associated with the requested URL, processor **111** forwards the URL to public network **100** via firewall **113**. Public network **100** returns the requested information to the identified user via firewall **113**, processor **111** and LAN **110**. Contrastingly, if a URL is included in a resource rating category for which the requesting user is not cleared, processor **111** denies the request for information.

In addition, the URL rating data within the above described systems can include a text listing of the rationale upon which a given rating is based, or additional information that facilitates more complex conditional rating schemes. As an illustration of a conditional rating for a URL assume that a the resource rating associated with a particular URL has been rated V for violent, and that all the terminals within a given school have clearances of NV (no violence). Therefore, in general, none of the school terminals would be granted use of the V rated URL. However, situations could arise that require exception to this general rule. For example, a certain terminal associated with a history class could need to access a particular resource that contained violent, but relevant information on an historic military battle. To facilitate access to such resources, the relational database rating information for the military battle resource would be augmented to reflect the conditional rating of "NV for user terminals located in history classrooms; V for all other terminals". With this conditional system, history class terminals would be restricted from all other "violent" rated URLs, but still be capable of accessing historically significant, yet violent, network resources. Conditional access could also be granted to terminals or users a function of time (i.e.; access limited to certain times of day for certain users or user terminals).

As stated above, the relational databases within the systems of FIG. **1** and FIG. **2** contain listings of user/user terminal identification codes and URLs. These listings are subjectively categorized or rated to facilitate the selective access of otherwise public network resources. This categorization/rating was assumed to have been performed by a system manager, and is effected by modifying the contents of the relational database utilized in practicing the invention. Within the system shown in FIG. **3**, processor **111** can be programmed to allow resource categorization information (listing **300**) and/or user/user terminal clearance information (listing **301**) within relational database **302** to be modified only by a specific dedicated management terminal **303**. Restricting ability to "write" new information into relational database **302** to management terminal **303** minimizes opportunities for database tampering. Alternately, the system can also be configured to permit database modification to be performed from any one of user terminals **107**, **108** or **109**. To protect against corruption of the contents of

relational database **302**, authorization for altering the contents of relational database **302** from a user terminal is controlled via use of a manager identifier. For example, if a system manager wished to modify relational database **302** from user terminal **108**, he or she would enter a password identifying themselves as an authorized system manager. The password is received by processor **111** and compared with the contents of manager ID memory listing **304**. If the received manager ID password corresponds to one stored in listing **304**, then user terminal **108** is identified as a manager terminal (as indicated by ID_{108} being stored within listing **304**). Modifications to the contents of relational database **302** may then be effected from that user terminal. When all modifications have been completed, the manager logs off and user terminal **108** returns to standard user terminal status (i.e., ID_{108} is cleared from listing **304**).

With the ever increasing proliferation of information systems in home, school and work environments, it is often the case that the responsibility of managing information access falls upon one or more individuals that are less than expert with respect to computer or information systems. Any of the above described systems can be implemented in a manner that allows a non-expert manager to easily control the systems. For example, within the system of FIG. **3**, processor **111** can be programmed to provide users recognized as system managers with an HTML "rating header" prior to the lead page of each retrieved network resource. If a manager retrieved the AT&T 800 Directory network resource via public network **100**, the returned information would be labeled by processor **111** to reflect a non-violent rating (see FIG. **4**, note the "NV" designation that precedes the retrieved resource—the AT&T 800 Directory). The manager may review the reasoning behind the rating by clicking on the portion of the HTML rating page labeled "click here". This results in the retrieval from resource categorization information listing **300** of the rationale upon which the NV rating was based (see the page shown in FIG. **5**). If the manager wished to disagree with the assigned rating upon retrieving the AT&T 800 Directory resource, he or she would click on "If you disagree, click here". This retrieves rating and rationale information from resource categorization information listing **300**, and provides the manager with a page that facilitates editing of the rating (see FIG. **6**). This page provides the manager with the current rating of the resource ("NV"), the main reason it was rated as such ("zero violent content"), and an area for entering a more detailed reason ("The resource consists of telephone listings . . . "). Upon completing, or modifying this HTML page, the system manager would select "Send Message" and thereby transmit the page to relational database **302** for storage within listing **300**.

It will be understood that the particular system and method described above is only illustrative of the principles of the present invention, and that various modifications could be made by those skilled in the art without departing from the scope and spirit of the present invention, which is limited only by the claims that follow. For example, any one of the above described embodiments could be modified to accept requests from users/user terminals that are in a format other than a URL. The relational database would merely have to be modified to store sets of information indicative of the particular type of request format being employed, and associated with a particular user class. Yet another modification would involve the adaptation to a multi-manager environment. In such an environment, network resource ratings could be arrived at as a result of voting among a number of system managers. For example, a number of managers could submit or alter a resource's rating, but the ultimate rating stored in the relation database would be an averaging of the submitted ratings, or whatever the majority of the managers chose as the rating of the particular resource. The relational database utilized in systems facilitating the invention could also be configured so that information indicative of allowable resource access is arranged to conform to resources that are configured in a tree structure format (such as a hierarchical directory arrangement). Such a relational database would include a listing of directory and/or subdirectory identifiers that could be labeled with a particular resource rating. The system could be configured so that resources located within a directory or subdirectory so labeled, would assume the rating of the overall directory/subdirectory. Alternatively, the system could employ a prioritized directory/subdirectory rating system. In such a system, a directory would be assigned an overall rating such as "NV". Particular items or subdirectories within this NV rated directory could then be labeled with specific ratings outside of "NV", such as "V". When a user accessed the NV rated directory, all items within it would be assumed to have an NV rating, except those items or subdirectories labeled with some other, more specific and different rating.

The invention claimed is:

1. A system for selectively restricting access to one or more otherwise public information resources, comprising:

a relational database containing a first stored listing that associates each of a plurality of resource identifiers with at least one resource rating, and a second stored listing that associates each of a plurality of user identification codes with at least one user clearance rating;

a processor adapted to receive a request for network access to one or more particular network resources, said request including a resource identifier and a user identification code, said processor being further adapted to query said first and second listings within said relational database, and execute said request for network access to said one or more particular network resources as a function of the resource rating shown to be associated with said received resource identifier within said first listing, and the user clearance rating shown to be associated with said received user identification code within said second listing.

2. The invention of claim **1** wherein at least one of said one or more particular network resources includes at least one image.

3. The invention of claim **1** wherein said processor is programmed to execute said request for access if said resource rating associated with said received resource identifier within said first listing, corresponds to at least one of said user clearance ratings associated with said received user identification code within said second listing.

4. The invention of claim **1** wherein said processor is programmed to deny execution of said request for access if said resource rating associated with said received resource identifier within said first listing, corresponds to at least one of said user clearance ratings associated with said received user identification code within said second listing.

5. The invention of claim **1** wherein said processor is contained within a network proxy server.

6. The invention of claim **1** wherein each of said user identification codes identifies one or more terminals adapted for facilitating network access to one or more particular network resources.

7. The invention of claim **1** wherein each of said user identification codes identifies one or more individuals authorized to access one or more particular network resources.

8. The invention of claim 1 wherein each of said resource identifiers corresponds to one or more uniform resource locators for accessing one or more particular network resources.

9. The invention of claim 1 wherein said relational database further includes a data listing associated with one or more of said plurality of resource identifiers, wherein said data listing represents textual information related to the resource rating shown to be associated with said one or more of said plurality of resource identifiers within said first listing.

10. The invention of claim 1 wherein said relational database further includes a conditional data listing associated with one or more of said resource identifiers, wherein said conditional data listing represents information indicative of specific conditions under which requests for network access to particular network resources associated with said resource identifier can be executed, and wherein said processor is further adapted to execute said request for network access to said one or more particular network resources as a function of said conditional data listing.

11. The invention of claim 1 wherein said relational database further comprises a stored listing of at least one system manager identifier, and said processor is adapted to identify a user as a system manager on the basis of said system manager identifier listing, and thereby permit said identified system manager to modify the contents said relational database.

12. The invention of claim 11 wherein said relational database further comprises a stored listing containing at least one HTML page adapted to facilitate the modification of the contents of said relational database by said identified system manager.

13. A method for selectively restricting access to one or more otherwise public information resources, comprising the steps of:

receiving a request for access to one or more particular information resources, wherein said request includes a user identification code and a resource identifier;

comparing said received request for access to a relational database containing a stored listing of user identification codes and resource identifiers, wherein each of said resource identifiers is associated with at least one resource rating, and wherein each of said user identification codes is associated with at least one user clearance rating;

executing said request for access as a function of the resource rating shown to be associated with said received resource identifier within said stored listing, and the user clearance rating shown to be associated with said received user identification code within said stored listing.

14. The method of claim 13 wherein at least one of said one or more particular network resources includes at least one image.

15. The method of claim 13 wherein the execution of said request for access is performed if said stored listing shows said received user identification code to be associated with at least one user clearance corresponding to at least one resource rating shown to be associated with said one or more particular network resources.

16. The method of claim 13 wherein the execution of said request for access is denied if said stored listing shows said received user identification code to be associated with at least one user clearance corresponding to at least one resource rating shown to be associated with said one or more particular network resources.

17. The method of claim 13 wherein each of said user identification codes identifies one or more terminals adapted for facilitating network access to one or more particular network resources.

18. The method of claim 13 wherein each of said user identification codes identifies one or more individuals authorized to access one or more particular network resources.

19. The method of claim 13 wherein each of said resource identifiers corresponds to one or more uniform resource locators for accessing said one or more particular network resources.

20. The method of claim 13 further comprising the step of providing a user with access to a data listing within said relational database, wherein said data listing is associated with one or more of said plurality of resource identifiers, and wherein said data listing represents textual information related to the resource rating shown to be associated with said one or more of said plurality of resource identifiers within said stored listing.

21. The method of claim 13 wherein said relational database further comprises a stored listing of at least one system manager identifier, and said processor is adapted to identify a user as a system manager on the basis of said system manager identifier listing, and thereby permit said identified system manager to modify the contents said relational database.

22. The invention of claim 1 wherein said plurality of resource identifiers associated with at least one resource rating are arranged in a hierarchical directory data structure.

23. The invention of claim 22 wherein said plurality of resource identifiers arranged in said hierarchical directory data structure are associated with more than one resource rating.

* * * * *

APPENDIX E

(U.S. PATENT NO. 5,954,640, "SZABO")

US005954640A

United States Patent [19]

Szabo

[11] Patent Number: 5,954,640

[45] Date of Patent: Sep. 21, 1999

[54] **NUTRITIONAL OPTIMIZATION METHOD**

[76] Inventor: **Andrew J. Szabo**, 130 Washington St., Dobbs Ferry, N.Y. 10522

[21] Appl. No.: **08/671,413**

[22] Filed: **Jun. 27, 1996**

[51] **Int. Cl.**[6] **A61B 5/00**

[52] **U.S. Cl.** **600/300**; 128/921

[58] **Field of Search** 600/300, 301; 128/920, 921, 923, 924

[56] **References Cited**

U.S. PATENT DOCUMENTS

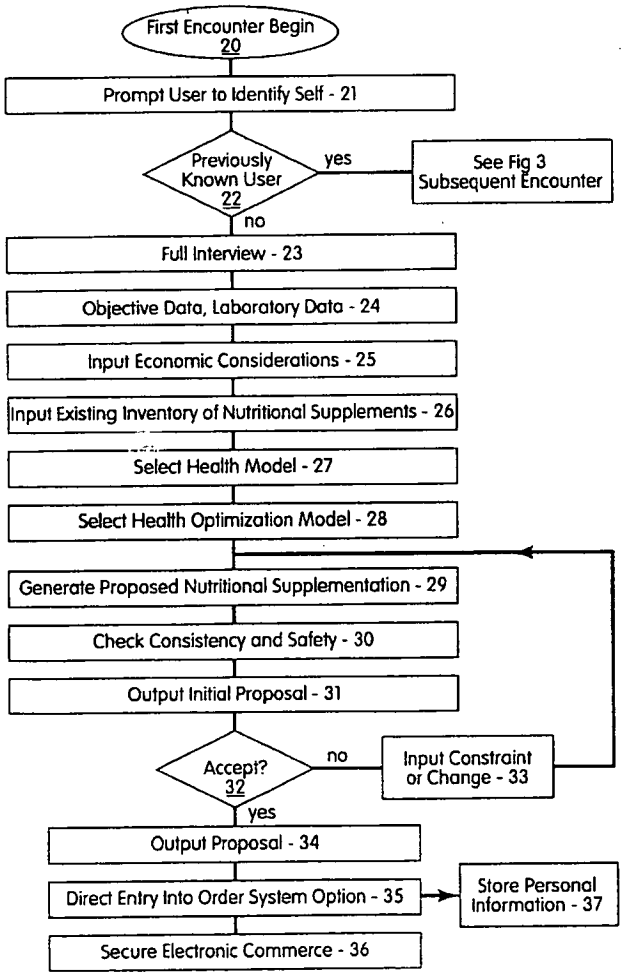
4,731,726	3/1988	Allen, III	600/300
4,954,954	9/1990	Madsen et al. .	
5,233,520	8/1993	Kretsch et al. .	
5,412,560	5/1995	Dennision .	
5,412,564	5/1995	Ecer	128/921
5,692,501	12/1997	Minturn	600/301
5,836,312	11/1998	Moore	128/921

Primary Examiner—Linda C. M. Dvorak
Assistant Examiner—Bryan K. Yarnell
Attorney, Agent, or Firm—Milde, Hoffberg & Macklin, LLP

[57] **ABSTRACT**

A method for proposing and providing nutritional supplementation for a person comprising the steps of receiving personal information, e.g., relating to health and diet, about the person, determining a health model for the person, determining an effect on the health model for at least two nutritional supplements, optimizing a proposed nutritional supplementation for the person based on the personal information about the person and effect for the at least two nutritional supplements, through employment of the health model, and outputting a proposed nutritional supplementation including amounts of at least two nutritional supplements. The method may also receive economic considerations, e.g., a budget, for the nutritional supplementation, and further optimize the nutritional supplementation based on the economic considerations.

41 Claims, 3 Drawing Sheets



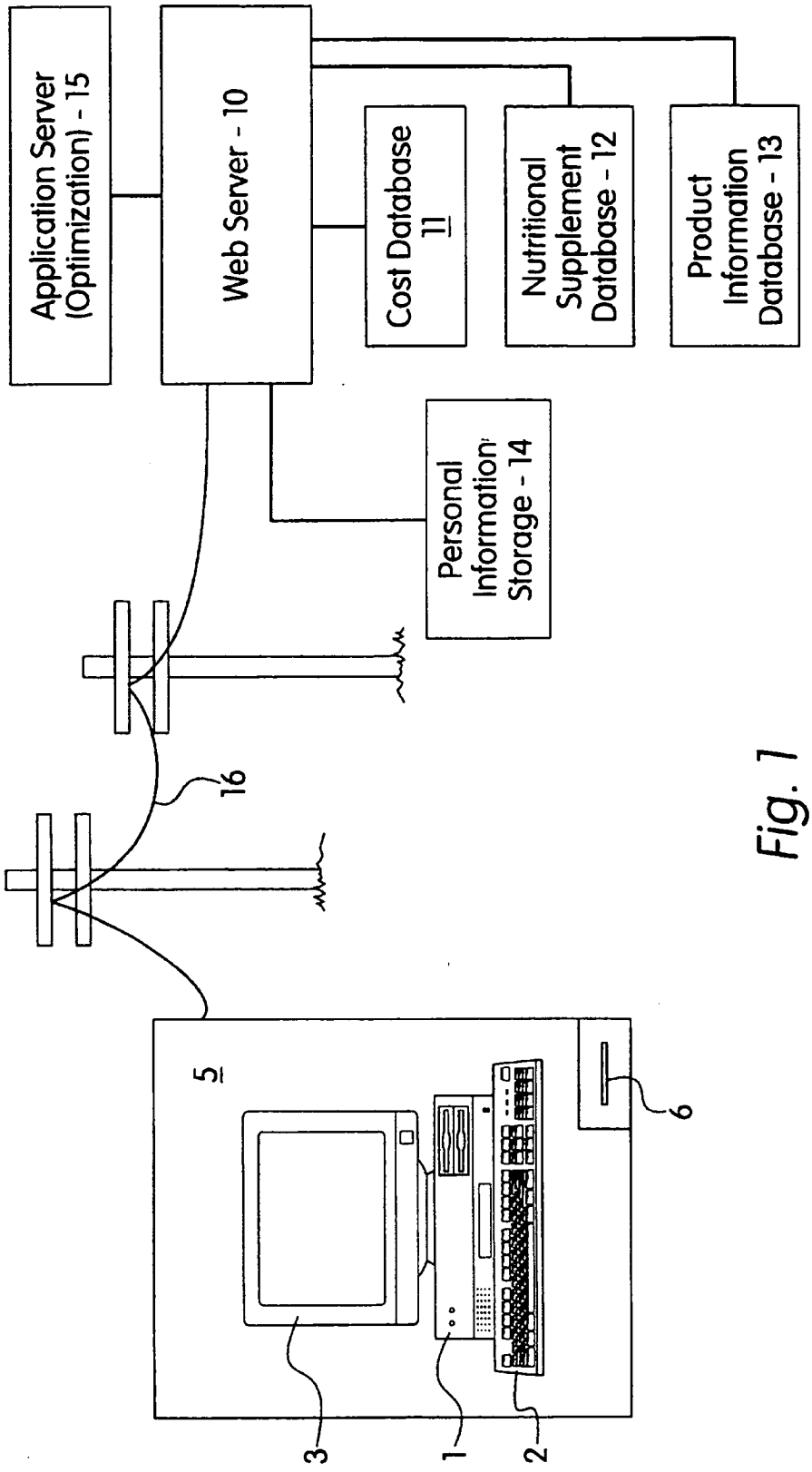

Application Server (Optimization) - 15
Web Server - 10
Cost Database 11
Nutritional Supplement Database - 12
Product Information Database - 13
Personal Information Storage - 14
16
5
6
3
1
2

Fig. 1

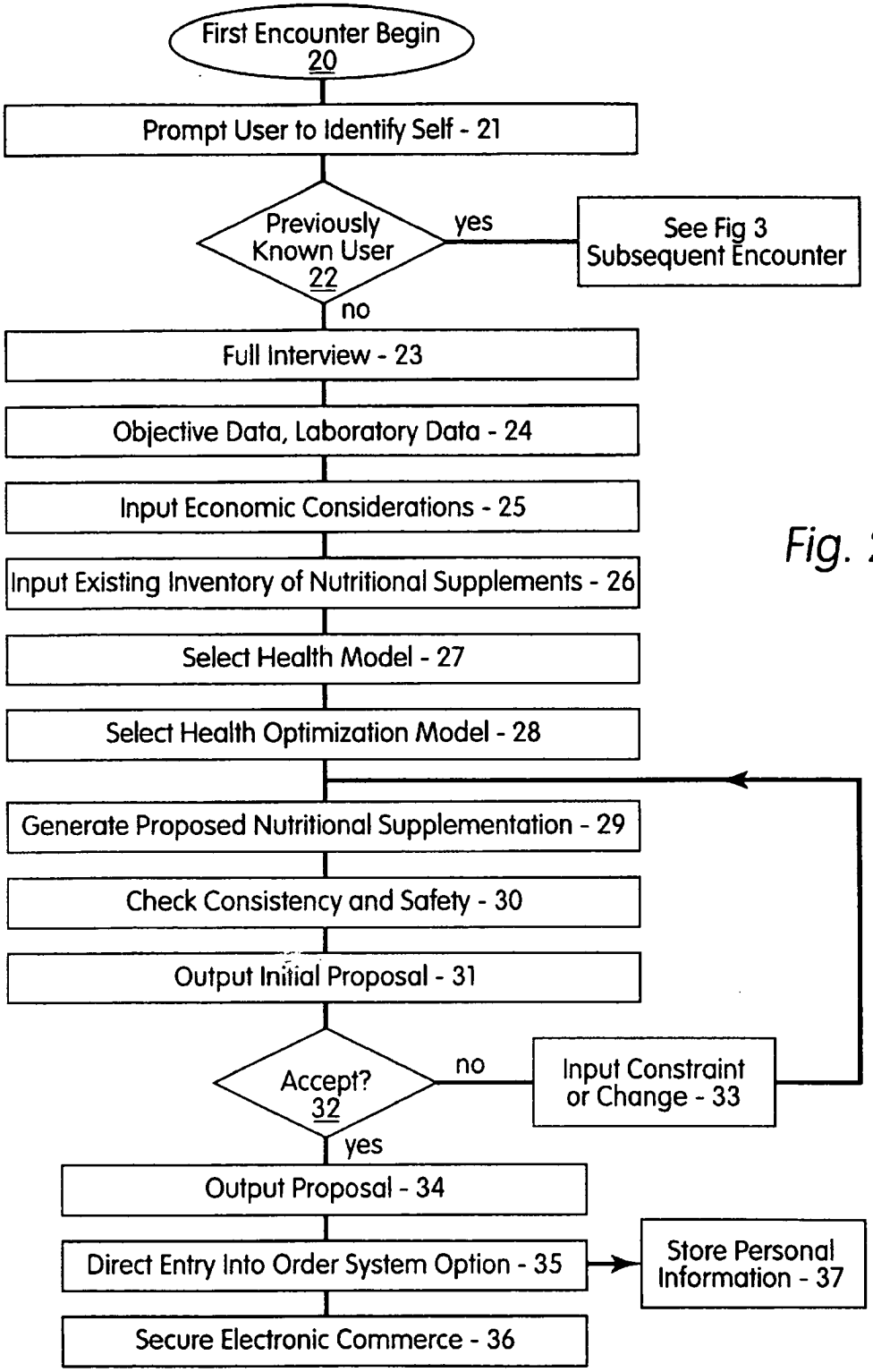

First Encounter Begin
20
Prompt User to Identify Self - 21
Previously Known User
22
yes
See Fig 3 Subsequent Encounter
no
Full Interview - 23
Objective Data, Laboratory Data - 24
Input Economic Considerations - 25
Input Existing Inventory of Nutritional Supplements - 26
Select Health Model - 27
Select Health Optimization Model - 28
Generate Proposed Nutritional Supplementation - 29
Check Consistency and Safety - 30
Output Initial Proposal - 31
Accept?
32
no
Input Constraint or Change - 33
yes
Output Proposal - 34
Direct Entry Into Order System Option - 35
Store Personal Information - 37
Secure Electronic Commerce - 36

Fig. 2

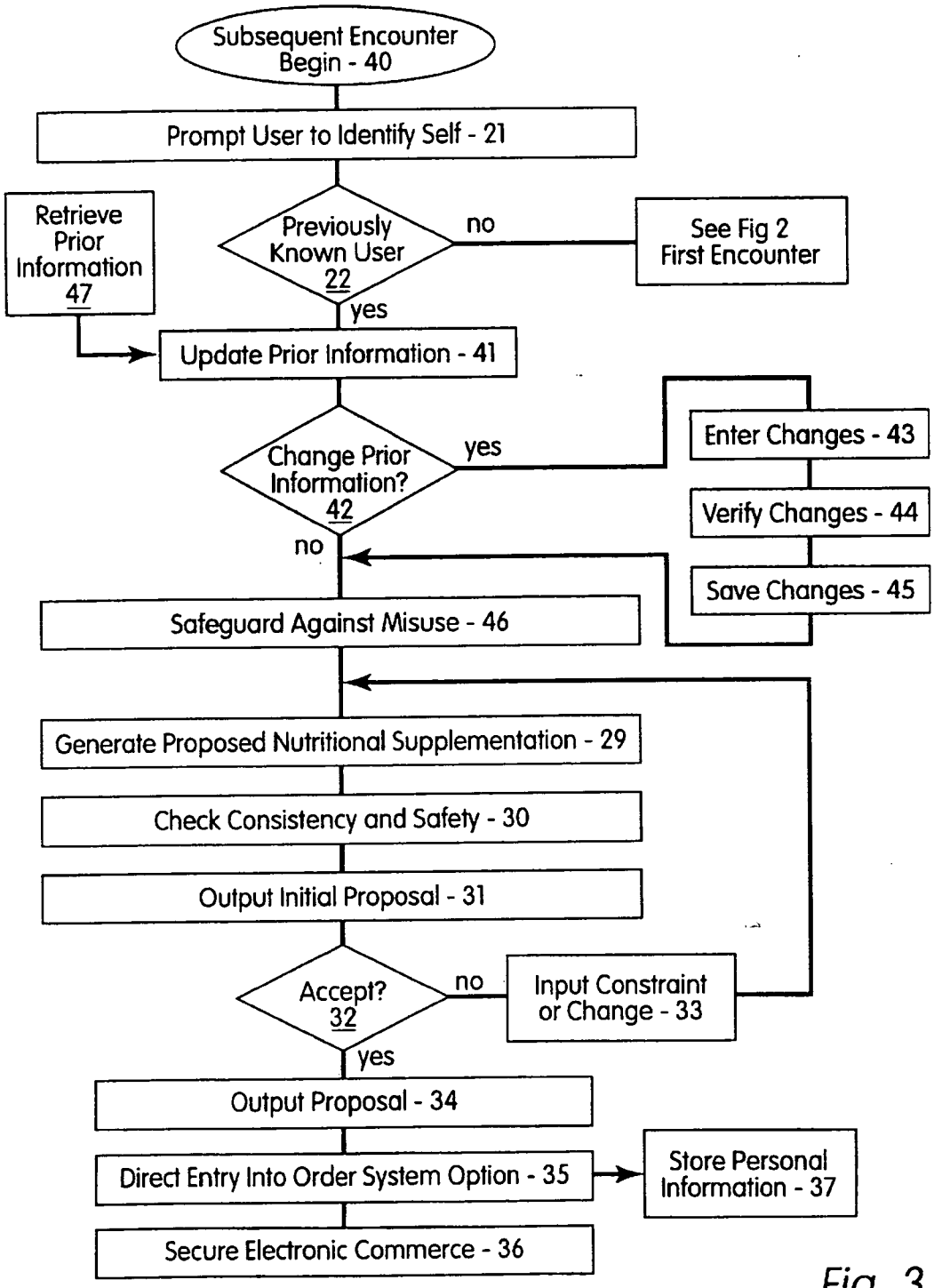

Subsequent Encounter Begin - 40
Prompt User to Identify Self - 21
Retrieve Prior Information 47
Previously Known User 22
no
See Fig 2 First Encounter
yes
Update Prior Information - 41
Change Prior Information? 42
yes
Enter Changes - 43
Verify Changes - 44
Save Changes - 45
no
Safeguard Against Misuse - 46
Generate Proposed Nutritional Supplementation - 29
Check Consistency and Safety - 30
Output Initial Proposal - 31
Accept? 32
no
Input Constraint or Change - 33
yes
Output Proposal - 34
Direct Entry Into Order System Option - 35
Store Personal Information - 37
Secure Electronic Commerce - 36

Fig. 3

NUTRITIONAL OPTIMIZATION METHOD

FIELD OF THE INVENTION

The present invention relates to the field of nutritional optimization systems and methods, as well as to apparatus for the analysis, storage and retrieval of nutritional optimization data and a system for delivering a nutritionally optimized product.

BACKGROUND OF THE INVENTION

It is well known in the art to analyze a diet of a mammal for typical macronutrients, such as carbohydrates, fiber, fat, protein, major vitamins and minerals. In fact, there is a well developed field of nutrition which seeks to employ medical and public health principles to determine an "optimal" diet. Further, total parenteral nutrition is known, wherein foods are determined and administered to a patient, often in a controlled environment. Infant formulas are also a known area of economic and dietary optimization based on public health and medical considerations.

Multivitamins are known, wherein a mixture of vitamins, minerals and cofactors are provided in a convenient dosage form. The levels of components are generally selected to be a significant portion of a recommended daily allowance (RDA) and up to about ten times the RDA. These multivitamins, however, are available in limited varieties, e.g., children's, women's, men's, and senior citizen's.

Of particular import in these many known systems is that these systems are not open to conjectural nutritional effects of components. Thus, if the effect of a component is not specifically known, these systems have no way to scientifically analyze its inclusion in a proposed optimized diet.

A known system, disclosed in U.S. Pat. No. 5,478,989, incorporated herein by reference, provides a computerized shopping cart for a supermarket which includes a bar code scanner, allowing typical UPC codes on food packaging to be read. A database of information about the food item may then be recalled, which may include labeling information. The consumer inputs personal information, and the retrieved information is presented in relation to the personal information of the consumer. This system, however, does not include an economic model, and does not relate to nutritional supplements for which no medical benefits are claimed. Further, this system does not make proposals, but rather returns processed information based on an input representing an item and the personal information.

SUMMARY OF THE INVENTION

The present invention provides an optimization of nutritional supplementation based on models that allow prediction of a change in health from an existing status, as a result of administration of a plurality of nutritional supplements. Relevant to various embodiments of the invention are activity of each nutritional supplement, desired change in status, toxicity and adverse effects of nutritional supplements, interactions between nutritional supplements and other factors, cost and economics of the nutritional supplementation, and risk, both positive and negative.

The present invention addresses the issue of nutritional supplements of incompletely or equivocally known value, and is thus not limited to predictions of the effects of unequivocally proven medical effects of supplements. In general, claims of medical benefit are not made for nutritional supplements, with the exception of known vitamins, minerals and bioavailable cofactors, and there is little standardization for dosage and regimens. As such, the database as to each of the factors in the optimization, with the exception of cost, may be partially undetermined. Thus, in contrast to known systems which operate on concrete data and established and accepted principles, the system according to the present invention operates to propose an optimization with incomplete or inconsistent data.

Another embodiment of the present invention employs a model of health of a mammal which encompasses both traditional nutritional analysis as well as unverified benefits of nutritional supplements for which no official or universally recognized standards are established. In fact, each individual may select a particular health model to employ, which may be different from or even partially inconsistent with other available or known health models. Thus, the individual is allowed personal choice of the model selected in the optimization.

The present invention also allows an optimization of nutrition and nutritional supplementation for a group of persons, such as a family. Thus, the optimization of the group health proceeds similarly to a public health analysis, e.g., the maximum good for the greatest number, while preventing detriment to the individual. However, this model operates on defined individuals, who preferably each have a health and personal information database record. The optional economic model, therefore, operates on the larger group rather than the individual person. In an analogous manner, an optimization may be created by the present invention for any delimited group, where that group's characteristics may differ from that of the "population" contemplated by a public health model. For example, a group of scouts on a camping trip will generally be expected to have similar activity and exercise levels, similar age and fitness, and therefore the menu for the group may be optimized based on a budget and health model. Parallel considerations would apply to food service in such institutions as school cafeterias, prisons, hospitals, welfare kitchens, or workplaces. The methods and apparatus according to the present invention allow particularized nutritional supplementation of the individual or group, to achieve an optimum health benefit. As noted below, the system may be integrated with various apparatuses to assist a consumer in shopping for foodstuffs and nutritional supplements.

A preferred embodiment of the invention employs an economic optimization of nutritional supplementation. Therefore, in addition to determining which nutritional supplements are appropriate, the cost of each component or the proposed nutritional supplementation as a whole is determined and used to achieve the maximum health benefit for given economic factors, such as a budget. Therefore, as a further aspect of this embodiment, the cost structure of combination supplements and quantity discounts are considered. In addition, third party health insurers or life insurers may provide payments, discounts or rebates for the proposed regimen. Where an economic model is not explicitly employed, a user may be presented with one or more proposals having differing nutritional supplement costs, which may then be selected by the user.

While the present invention encompasses the tenets of non-traditional medicine and nutritional supplementation, it does not eschew traditional health schemes. Thus, known diagnostic, analytic and prognostic indicators can be employed to determine an optimization of nutritional supplementation for a given individual. Further, certain patients are fragile, and therefore a risk of health deterioration due to supposed nutritional health optimization is considered. Therefore, an aggressive health optimization

may be proposed for a healthy young individual, while a conservative approach may be proposed for an elderly patient with various ailments.

Further, the present optimization may be cognizant of medical, surgical or pharmaceutical treatments of a patient, as well as natural conditions such as menstruation or lactation, or disease, and determines or predicts any potential interactions between prescribed care and proposed supplementation in order to avoid adverse interactions or detrimental effect on the treatment regimen. Beneficial interactions are also cognized, and thus may be used to increase efficacy or efficiency.

Further, the present optimization may be cognizant of toxicity of the entire dietary and supplementary regimen, particularly in relation to the liver and kidneys, but also considering other organs and systems which may be stressed, including the heart, reproductive system and endocrine system.

The present invention may also provide a temporal optimization of nutritional supplementation, wherein diurnal, weekly, monthly and/or seasonal or life-cycle variations are considered and factored into the optimization scheme. Further, as a part of the cost optimization, dosing schedules and component half-lives may be considered in order to allow the most effective and most convenient nutritional supplementation. For example, the cost of a nutritional supplement generally is not solely related to the amount of a nutritional component in the supplement, and higher doses are generally less costly per unit than lower doses. On the other hand, higher purity components may be more expensive per unit dose, for example where the purification process is difficult. Such higher purity dosage forms may be desirable, for example where the impurities are harmful or have undesired effects, or where the shear volume of nutritional supplement is undesired. Thus, the proposed nutritional supplementation may include an analysis of dosage forms.

As used herein, nutritional supplements include foods, capsules, pills, powders, gums, and liquids, or other oral dosage forms which include known or quantifiable nutrients. Also encompassed are nutritional supplements delivered in any manner to the digestive system or intravenously, as well as nutritional supplements which are administered through other routes, such as through mucous membranes.

Often, nutritional supplements are provided in a form which includes excipients, impurities, or other components than the denominated nutritional supplement component. Therefore, another embodiment of the present invention analyzes, to the extent possible, the nature of the nutritional impurities or excipients, and include these components in the nutritional optimization.

In general, as stated above, macronutrient (foodstuff) optimizations are known, and the present invention may encompass this aspect of nutritional optimization as well. Thus, for example, an individual indicates his normal nutritional intake as an input to the system, which is to be supplemented or modified. The result of the optimization may therefore include a proposal to reduce intake of a supplement, macronutrient or foodstuff, as well as increasing or adding nutritional supplements. Thus, heavy consumption of milk may suggest lesser supplementation with fat soluble vitamins D and E, as well as calcium.

The present system provides an individually tailored proposal for nutritional supplementation or modification of intake. Being a proposal, and given the nature of mandates of dietary intake, the proposal may be accepted or rejected by the individual. Therefore, another embodiment of the invention involves an interactive process for arriving at a proposal, as well as a correction of optimization based on a deviation from a proposal. In this case, the cost optimization and risk analysis potentially play an important roles in a statistical analysis to arrive at a proposal. Since it would be expected that, except in the case of total parenteral nutrition, no absolute dietary schedule will be maintained, and further that it is primarily those individuals whose diets are most aberrant initially who are recalcitrant to change, the optimization proposal must include leeway for deviations.

Therefore, one embodiment of the invention provides an immediate feedback of a proposed nutritional supplementation based on an actual present status of a person, including recent meals and nutritional supplements, activity, health status and prospective events. This optimization may be provided through a hand held, pocket or bracelet (watch-type) device, personal computer, personal digital assistant (PDA), as a device which might be attached to or integral with a shopping cart, terminal to an on-line service, through the Internet (e.g., through a server or as a Java application), telephone with voice communication, kiosk, or centralized computer system. Therefore, a full featured system may be used to define an optimization, which may then be used to download an optimization to a portable or remote device. The programmed optimization may then be used to help keep the person "on track", and to report on an actual pattern of activity, diet and nutritional supplementation. While the portable or remote device may alter or reselect optimization continuously or often, preferably the optimization is performed infrequently, such as once per month.

Thus, a reoptimization may be performed periodically, e.g., monthly, or frequently, e.g., daily. The optimization procedure may also be provided as major optimizations, in which substantial changes to underlying models are implemented, and minor optimizations, where perturbations from a desired health status are corrected by nutritional supplementation according to a determined model.

A preferred embodiment includes an economic optimization because, without this factor playing an explicit role, the "more is better" theory may produce a proposal which is untenable. Known systems which attempt to optimize nutrition perform economic optimization in one of two ways. First, the public health model selects cost levels designed to do the most good for the most people. Some persons will receive a suboptimal dose, while others will receive little incremental benefit or even suffer toxic effects. Further, some persons will be asked to spend more than a reasonable amount, while others will have excess disposable funds without guidance as to how these funds should best be employed. Thus, the public health model does not account for an individual and his own specific factors, including budget. Second, an incomplete or limited economic analysis may be performed without the benefit of a linked health model. For example, an individual who visits a health food store and selects supplements performs a limited economic model, e.g., "that costs too much", in the selection of items for purchase. By linking the economic model with an individual health model, the benefits of a personalized proposal at acceptable cost is obtained. Further, by allowing a statistical error in the actual diet as compared to the proposed diet, the optimization may produce a better "real-world" result.

In operation, the system first obtains personal data about an individual. This may be obtained through automated data analysis, interview, survey, subjective analysis, laboratory testing, and the like. A database is provided with information

about available nutritional supplements, including contents, price, and dosage form. A further database includes information, including risks and benefits, about constituents of nutritional supplements. A system, preferably a computerized algorithm, computes a health model of the individual based on the input information, as well as a desired budget. This model computes a present state of health, according to the available information, and determines a desired state of health, based on the maximum benefit for the available funds and the available nutritional supplements.

The resulting nutritional supplements, intended to help a mammal reach the desired state, along with suggested changes in the existing diet, comprise the proposal. In appropriate circumstances, activity and exercise may also be aspects of the proposal. The individual, however, need not accept the proposal, and may thus interact with the system to modify the proposal in specific aspects. These changes act as constraints for a secondary modification of the proposal. For example, a selected health model may suggest 300 mg of ascorbic acid (vitamin C) per day, in three doses. However, the individual may prefer 750 mg per day in three doses. Thus, the proposal is then updated with 750 mg per day in three divided doses as a constraint. The entire health model must be recomputed based on this constraint. In recomputing the model, the system further determines whether this constraint implies that a different model is more appropriate for implementation. In order to resolve this issue, the individual may be queried to determine the reason for the preference. If appropriate, hybrid models may be employed. The nutritional supplement proposal may thus include timing and frequency of dosage of the nutritional supplementation.

In theory, an economic based model may result in a highly skewed proposal, with high doses of relatively cheap components and without any expensive components. However, often, temperance and variety are desired, and thus amounts of some nutritional supplements are limited and others added, even though these result in reduced benefits according to a strict scientific analysis. Thus, a perceived benefit of a nutritional supplement may be in excess of a rational analysis of the potential benefit based on a review of existing scientific data. Thus, a health model may include an analysis of a perceived benefit of a component, rather than necessarily a scientific analysis. Further, it is noted that, in accordance with the scientific method of analysis of nutritional supplementation, studies may fail to show a benefit, or produce contradictory findings, even for nutritional supplements of real value. For example, ginseng is believed by many to be beneficial, but many scientific studies have failed to reveal a health benefit. This does not mean, however, that the proposed benefit of a component is not real. Another limitation of scientific methods is that they emphasize dose-response relationships over balance. However, a perception of an individual may be that supplementation of smaller amounts of many different components is preferable to megadoses of a small number of nutritional supplements. Another limitation of typical scientific studies is a difficulty in proving subtle long-term effects of small doses.

A further embodiment of the implementation of the present invention includes an apparatus for formulating nutritional supplements. Thus, based on the proposal, custom formulation may be provided to an individual. Alternately, standard dosage forms may be selected for the proposal.

A still further embodiment of the invention provides a vending machine or point of sale dispensing machine which formulates or combines pre-prepared dosage forms of nutritional supplements based on the proposed nutritional supplementation. Where the point of sale dispensing machine is in a public location, a limited interface may be provided, for example, a touchscreen and a magnetic stripe or smart card interface. Thus, a person previously registered with a central system may present to a kiosk or free-standing machine, and be identified by a card, e.g., a credit card or smart card. The card is used to call up a record of the person, which is then employed to generate a "welcome" screen for the person. Such a machine can also provide for custom packaging of a group of standard dosage forms of nutritional supplements.

Optionally, a user may be interviewed by or in the presence of a trained professional, with the data inputted or accepted in an objectivized format. Thus, with a trained professional, e.g., a doctor, nurse, chiropractor, social worker or nutritionist, the input of medical information, analysis of choices, selection of models, and approval of proposals may be facilitated. The interaction between user and professional may also be part of a consultation or treatment session, and the data entry shared with a medical records system. Thus, the nutritional supplementation system may be integrated into traditional medical care settings, and users who are in need of traditional medical care directed away from potentially inappropriate self-help paradigms.

After initialization of the system, e.g., identification of the person, the person may then interact with the system, for example through a graphic user interface with a touchscreen input. The graphic user interface may employ standard constructs, such as menus, icons, and dialogue boxes. The screen interface may also be customizable for the user, e.g., language, level of sophistication, preferences, etc. In conjunction with the interface, a database retrieval system may be provided to assist the person in making choices and selections. Thus, a search engine may be accessed, as well as pre-formed strategies for searching various topics. Where the system connects to an on-line service, or the Internet, so-called "spiders" may be employed to retrieve information of a class specified by the user without requiring explicit identification of each record. Further, so-called agents may be used to assist in interaction with the system. The agents may include, for example, aspects of the various selected models. The software described may be executed on a server, client or stand-alone computer. In particular, the optimization may be performed on a personal computer, with the resulting proposal printed for manual delivery or transmitted to a host system. The present invention therefore envisions electronic commerce where an order executing the proposal is transmitted electronically, with the resulting goods delivered through standard channels, such as mails, couriers and parcel post. The present invention also envisions execution of the software at or near a point of sale, with the goods delivered at or in close proximity to the terminal. Thus, the terminal itself may include a vending machine or be in a retail nutritional supplement sales environment.

The system thus seeks to determine, based on a set of personal preferences and constraints, as well as a health model and optionally a personal economic optimization model, an optimal proposal for nutritional supplementation. Public health concerns partially defer to individual health considerations. Further, absolute health mandates defer, within limits, to personal preferences and optionally cost tolerance.

In a typical application, a consumer initially identifies himself to a computerized system and undergoes an interview process. If the consumer is known to the system, i.e.,

has a database record, the prior history of the consumer is recalled. Otherwise, the consumer is processed as a new user. The system may also have or be granted access to medical records of the user. Based on the interview, information relating to the consumer, including present health status, dietary habits, medical treatments, activities, exercise and preferences are determined. Further, a health theory is proposed, based on responses to particular questions or scenarios.

An economic model is optionally formulated, which may be as simple as a daily, weekly or monthly budget for nutritional supplements, or a more complicated analysis including normal food intake and expected health benefits. The economic model may also include expectation of third party benefits, such as payments, discounts, subsidies, rebates, insurance or copayment by health or life insurance organizations, health maintenance organizations, prepaid provider organizations, or others. In fact, these third parties may grant economic benefits which are dependent on a correspondence between an organizational health theory and a proposed health theory. Thus, the third party may skew the proposed nutritional supplementation based on selective economic benefits. It is also noted that economic constraints may change over time, and therefore a reoptimization may be required on each such change.

Based on an estimation of the present status of the consumer, the system then seeks to propose specific changes and nutritional supplements, in accordance with the health theory, expressed preferences, and optionally within the constraints of the economic model, to maximize the expected benefit to the consumer. The consumer then interacts with the system to "tune" the proposal based on personal preferences. After acceptance, the consumer may then execute the proposal by purchasing the recommended supplements. As stated above, the purchase system may be linked to the terminal, in communication with the terminal, or completely separate.

Over time, the system may determine whether the proposals are achieving a desired effect, to the extent that this is determinable. For example, medical tests or diagnoses, or subjective responses to inquiries, may be used as feedback data. Where appropriate, the system may be interfaced with diagnostic or exercise equipment, to obtain objective data. If the effect is as expected, then the proposal is reinforced. If the effect differs from the expected effect, then the proposal is reoptimized based on the feedback. Accordingly, if the consumer is a high responder to a nutritional supplement, the amount of the supplement may be increased as compared to other components. Alternately, the amount may be reduced, if the increased response is undesired or unnecessary. In an economically optimized system, economic resources may be freed for other nutritional supplements. Thus, the system may employ a closed loop feedback input with periodic reoptimization.

If a consumer alters his preference, or the health theory is altered, either by selection of a new theory by the consumer or an alteration in the theory based on new evidence, the subsequently generated proposals may also be altered. However, the system will continue to rely on closed loop feedback to personalize the proposals.

The personal interview will acquire data about the consumer's nutritional background, sex, weight, age, ethnic background and familial health risk factors, environmental and behavioral health risk factors, allergies, medical conditions, drugs currently being taken, treatments and responses, activities, exercise, as well as subjective factors. In order to further evaluate the consumer, it may also be desired to obtain data relating to diagnostic tests on the consumer, such as blood tests for levels of specific micronutrients or indicative of nutritional status.

In order for the economic optimization according to the preferred embodiment to be fully effective, a complete and accurate database of the costs of various options must be available. As such, one embodiment of the invention provides an electronically accessible database of nutritional supplement content and cost information. The database may also include information about the normal food budget of the consumer, since a change in the food budget may result in a change in the nutritional supplement budget.

The database of costs may be integrated with an inventory management system for a retail, wholesale or mail order nutritional supplement vendor, and may be accessed by, for example, by SKU or bar coded UPC symbols. Thus, while shopping, a user may be able to determine or test the effect of a particular proposed purchase on the optimization.

The health model or theory is a set of rules, formulae, statistics and factors which allow analysis of the present health status of the consumer as well as a hypothesized change in status due to one or more nutritional supplements. Linked to this health model are activity and toxicity models for the nutritional supplements, so that the type and amount of nutritional supplements to be proposed may be analyzed in conjunction with the present status of the consumer. In particular, the activity model proposes a benefit of a nutritional supplement, while a toxicity model compels a limitation in dose. The activity and toxicity models may be combined into an efficacy model. Where the individual models do not explicitly account for interactions with other factors, models, and nutritional supplements, a separate interaction model may be provided to inform the consumer of potential interactions and seek to prevent hazards or inefficiencies, and to determine whether beneficial interactions are present or may be increased, for example by combining magnesium and vitamin D. Another example is the ability of ascorbic acid to degrade nitrosamines, which form from nitrites in foods, for example preserved meats and smoked fish. Thus, the nutritional supplementation optimization may propose that orange juice, a food, be consumed when lox and bagels are also consumed. Thus, the proposal is not limited to nutritional supplementation with micronutrients alone.

The present system may also include a further related concept, a model for optimization of health, which differs from the health model by allowing statistical analysis of risks and benefits, as well as contingent benefits.

Thus, a number of models operate simultaneously to achieve a result, i.e., a proposal. First, the health model defines the status and proposed status of the consumer. Second, the efficacy models define a change in state with respect to amount of nutritional supplements. The efficacy models are separated from the health model to the extent desired so that each may be modified separately to include new information. Third, the optional economic model limits the optimization to affordable levels, and serves higher purpose as well. The implementation of an economic model facilitates economic efficiency by allowing providers to seek cost effective nutritional supplements, even if this effect is somewhat delayed. The optional optimization of health model seeks to compensate for statistical risk and benefit, which may be independent from the health model or efficacy models themselves.

While the optional economic model may be relatively static, this model may also be more dynamic, and include the

concepts of a "sale", discount coupon, incentives, quantity discount (individual component or gross order), handling, transaction costs and service charges, negotiations with the vendor, or other known economic perturbations or corrections. Further, as stated above, the economic model is subject to perturbation by the influences of third party payers. While the health model will generally not allow a third party to compel supplementation with an undesired nutrient, the economic model does weigh in favor of the subsidized nutritional supplements where these are beneficial.

In practice, these models are preferably provided as modular objects in a computer system, allowing one object to be substituted, altered or updated without simultaneously requiring consideration of corresponding or compensatory changes in other models which are not dependent on the changed object. Of course, the resulting optimization is a dependent object and must be recomputed after a change in a parent object. Each model therefore includes a set of formulae or parameters, which may be evaluated in context. The evaluation is a statistical or multifactorial optimization to determine a best proposal. As stated above, based on external inputs, factors of the model may be constrained. Further, closed loop feedback may be used to update or personalize the model for more accurate determinations.

The models or modeling system may also include neural networks or fuzzy logic paradigms. A neural network system is advantageous, for example, where a model may be expressed as a set of neural network weights, and therefore computing an optimization requires evaluation of the neural network. Preferably, the model is modular, with portions being separately evaluable and substitutable. Thus, the model may be formed of a set of modules, representing aspects to be optimized. A fuzzy logic system may be advantageous where semantic expressions may be used to describe a relationship, and where a precise logical statement of the relationship is difficult to determine or evaluate. A neural network is generally created by an iterative training process based on empirical data in a training process, while a fuzzy logic system is generated as a set of explicit rules in a programming process. Neural networks and fuzzy logic systems may be adaptive, i.e., changing over time based on feedback of an actual effect, to better predict how a nutritional supplementation may achieve a desired effect.

The system according to the present invention may also be employed with patients, i.e., persons under medical care for a disease. Thus, under such circumstances, the health model may particularly include a model of a disease, with the parameters of the proposal reviewed by a medical practitioner prior to implementation. Thus, for example, the system may be employed to optimize a total parenteral nutritional program for a patient. In this case, however, an economic optimization may be excluded or play a lesser role due to the high cost of medical interventions and the possible role of a hospital pharmacy or pharmacist. Even where total parenteral nutrition is not the desired result, a patient under medical care may benefit from the proposal. It is noted that where a medical professional makes the decisions, the health model will generally be conservative, i.e., employing accepted scientific theories and relationships, while the health optimization will also be conservative, e.g., "low risk". On the other hand, in some such cases, the efficacy models chosen may be very aggressive, in view of the medical supervision and the availability of close monitoring. Thus, the proposed doses may be closer to those at which toxic or adverse effects may be seen.

In forming the health models, as well as the other models, the necessary data need not be inputted as specific rules. Thus, the models themselves may be adaptive based on the experiences of individual users or groups of users. Therefore, an objective and subjective feedback to the system may be used to improve the models and allow the predictions based on the models to more accurately reflect true consumer experience. As stated above, neural network technology and other adaptive paradigms may be employed to dynamically improve the models though use and feedback. Preferably, the raw data from users are not used directly to update the models, because this may lead to anomalies and subjective skewing. Rather, a filter is applied, preferably in the form of a trained person, to review the feedback data to determine the nature of the adaptation to be implemented and to control the process. Automated systems may also be used. Alternately, the feedback may be separately analyzed, and used as a basis for allowing consumers to select a different model, which may be perceived as more appropriate. For example, if a consumer experiences an undesired side effect from a particular nutritional supplement, the consumer may review information relating to others who consume the specific nutritional supplement, or others who suffer the same undesired side effect. This will allow the consumer to benefit from the experiences of others to potentially allow a change in nutritional supplementation to avoid the side effect. As another example, where the optional economic model is not a fixed budget for a limited period, the actual experiences of users or the specific consumer, including missed doses, change in diet, weight loss or gain (potentially resulting in dose changes over time), change in body fat content (potentially resulting in desired changes in fat soluble or water soluble nutritional supplements), and other factors may be used to more accurately match the nutritional supplementation proposal to the economic limitations.

The proposal need not be limited to nutritional supplements, and therefore changes in diet, activity or exercise may also be included in the proposals. It is noted that great changes in diet, activity and exercise are difficult to effect, and therefore such proposals may be of limited benefit. In fact, since non-compliance rates are expected to be high, an optimization based on a proposal requiring distinct efforts is likely to be rejected or ignored. On the other hand, simple changes in diet, which are likely to be adopted, may be very efficacious. Thus, on a pragmatic basis, the proposal preferably emphasizes small dietary changes and a regimen of pills and/or supplements, even where an equivalent change might be possible through extensive dietary modification or restriction. The user may therefore weight a relative expected importance of normal diet as compared to nutritional supplements.

Once a proposal is accepted, the system preferably has a link to a system for ordering nutritional supplements recommended in the proposal in standard packaging. Thus, according to the preferred embodiment, the economic model database system may include a link to an on-line ordering system from a nutritional supplement supplier. Alternately, the nutritional supplements may be individually formulated for a consumer from standardized ingredients. The proposal, once accepted may be directly entered into an ordering system, transmitted through an on-line service, e-mailed, printed, or directly filled as an order. Alternately, various supplements may be convenience packaged by daily dose.

Where the transaction occurs through an electronic medium, e.g., the Internet, the payments may involve traditional means, such as credit cards, or may involve newer systems for electronic commerce. Where sensitive medical or accounting information is transmitted through a public

medium, it is preferably encrypted, such as with an RSA public key/private key algorithm, the PGP algorithm, or other encryption technique to prevent interception and ensure authenticity.

It is noted that often consumers will have a present regimen of nutritional supplements, possibly with existing inventory. In this case, assuming the consumer preference is to continue the existing regimen, the regimen will act as a constraint or soft constraint on the optimization, and the system will propose additional nutritional supplements and possibly modifications of the regimen. For example, where an existing regimen provides too much of a nutritional supplement according to a health model, the proposal may recommend lowering intake of that nutritional supplement. Where a constituent is inconsistent with the health model, the proposal will exclude that constituent. For example, a high iron intake might be inconsistent with an antioxidant health model, other things being equal. On the other hand, where an available constituent substitutes for an unavailable, though potentially preferable constituent, the optimization allows ingestion of an effective amount of the available constituent until the supply is exhausted, at which time the constraint is removed.

As an adjunct to selection of a health model, the system may also include educational features to inform the consumer about health, nutritional supplements in general, or specific nutritional supplements. Thus, a database entry may be provided for each nutritional supplement with both cost information as well as educational information. Further, a data retrieval system may be available to allow a consumer access to the information in a nonpredetermined manner. This educational information may also be used to guide the process of selecting a health model and identifying and analyzing risks for the health optimization model. The database may be provided locally or through an on-line or Internet based service. Further, searching capabilities may employ typical Internet searching techniques, for example to retrieve Usenet messages or world wide web pages.

Where educational information is provided, information may generally be segregated into two different categories. First, mainstream science published in peer reviewed journals represents a high quality source of information. However, such information may be delayed by the peer review process or the underlying studies may be prolonged. On the other hand, non-peer reviewed information and non-mainstream journals may develop hypotheses which require years of clinical testing in order to pass muster under the peer review process. Thus, while non-peer review information may be less trustworthy, it may be important and suggestive nevertheless. Further, the mainstream scientific community does not always address nutritional supplements in a timely manner, and therefore the non-peer review or non-mainstream publications may be the sole source of information or suggestion relating to certain types of nutritional supplements. The system may also be used to present topics of dissent and debate, which may form the essential differences between health models, and thereby present the distinctions and allow informed selection of a desired health model.

This distinction is also drawn elsewhere in the optional optimization process. The health optimization model factors in risk tolerance as a separate factor. Thus, a consumer with high risk tolerance might give greater emphasis to alternative medicine concepts than a lower risk tolerance consumer. This risk tolerance may be explicit, i.e., a person who expressly desires a higher risk (and higher potential reward) proposal, or implicit, i.e., a person who is healthy and can tolerate adverse effects better than an ill or fragile person.

On one hand, a Japanese user would likely find comfort in a traditional Japanese health model, which in western medicine is considered "alternate". On the other hand, an orthodox American medical practitioner using the nutritional supplement optimization system is unlikely to adopt substantial contributions from alternative medicine sources.

In order to avoid providing medical advice, which may permissibly be provided by a licensed physician, the system may instead provide access to the actual studies and works of authorship, without editorialization. Thus, the system may include a mass database of selected references which may inform the user of the risks and benefits of a given nutritional supplement, as well as aspects of a health model. Likewise, where official government advisory information exists, this information may be presented to the consumer. For example, alcoholic beverages in various forms may be considered beneficial in moderate amounts, yet toxic and addictive in higher doses, and pose substantial risks while driving or operating machinery. Therefore, alcohol products are accompanied by a warning. This warning, as well as various studies which support the use or abstention from use, may be available for review by the consumer. Thus, an herbal extract in alcohol may include a significant dose of alcohol with an effective dose of the herbal ingredient. Whether this alcohol is considered toxic or beneficial may depend on the health model chosen and also the contemplated activities of the consumer.

The interface to the system is preferably an interactive graphic user interface, allowing the consumer to make incremental selections and make modifications to selections during a session. The use of screen buttons, hot links, menus, dialogue boxes and other typical graphic user interface elements is therefore preferred. Through use of the system, the preferences of the user may be determined, and present further data and selections based on the determined preferences. Such a learning interface may allow efficient interaction between the machine and user. The system may be, for example, a Pentium® personal computer, Apple Power PC®, UNIX system, or other known type of computer system. The operating system is, for example, Windows for Workgroups 3.11, Windows 95, Windows NT, Macintosh Operating System, SunOS, Netscape/Java, or other known type.

While the various models seek to optimize health under the various constraints and inputs, a separate function is preferably provided to confirm that potentially dangerous or undesirable amounts or combinations are not proposed or selected. For example, excess amounts of fat soluble vitamins are to be avoided, and the calculations should encompass both the nutritional supplementation as well as the dietary load.

Preferably a proposal is accompanied by a statement or warning of potential or common side effects or adverse effects relating to the components of the proposal and the dosages and dosage forms proposed. Thus, a tailored screen output or printout specific for the user, which may specifically refer to the user's medical history or susceptibilities, is generated. During an encounter with the system, the user may be interviewed for serious or common effects, and warned to seek medical attention if an effect so warrants.

The system may reside on a local computer, network, client-server environment, on-line service, the Internet, or in other types of environments. Preferably, the consumer interacts with a terminal which has access to a remote database which includes model components, so that each terminal need not include the entire database. However, it is also

possible to load the databases in a storage medium, e.g., CD-ROMs, so that a computer network is not required. These CD-ROMs would likely require periodic updating, for example, with changing economic and scientific information. The product information and economic information database together comprise a form of nutritional supplement catalog with current market prices, from which orders may be reliably produced.

BRIEF DESCRIPTION OF THE DRAWINGS

The preferred embodiments of the invention will now be described with respect to the figures, in which:

FIG. 1 shows a schematic view of a client server system for interacting with the optimization system according to the present invention;

FIG. 2 shows a flow chart for processing a new user according to the present invention; and

FIG. 3 shows a flow chart for processing an experienced user according to the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

A computer system is provided in a client-server environment. As shown in FIG. **1**, a client system **1** includes a human interface, having a keyboard **2** or human voice auditory input, touchscreen **3**, and video display **4** in a kiosk **5**. The client computer is a Pentium® PC running Windows 95 and Netscape 3.0/Java. A hardcopy output device **6** (hidden in the drawing) is also present, e.g., a printer. The server, which is a Sun web server **10**, stores the databases for cost **11** and nutritional supplement information **12**, as well as the modeling information **13**, and includes an application server **15** to evaluate the models and generate a proposal. Personal information is stored in a storage system **14**, including health profiles, personal preferences, selected models, and other pertinent data. The client and server are linked by a network **16**, e.g., TCP/IP over PPP dial up lines through the Internet, or over Ethernet, ISDN, Frame Relay, or other known means. The server **10** may also be linked to other systems, e.g., for deriving demographic information about registered persons for, e.g., targeted marketing.

As shown in FIG. **2**, the user is prompted to input various information, first identifying himself **21**. If the user is previously known to the system **22**, the prior data is retrieved **47** from the personal information storage **14** and employed, and the user need only update information **41**, including any change in health status, diet, non-compliance with prior proposal, weight, diagnostic tests, preferences, or economic constraints. Of course, the user may also edit or change any prior information **42**, which is entered **43**, then verified **44**, and finally saved **45**. However, safeguards are placed to prevent intentional deception of the system to force an unsafe or unwise proposal **46**, by placing risk limits and noting unexpected changes in data. Safeguards are also placed to prevent unauthorized intrusion into an individual's personal information records.

Where a novice user first uses the system, a full interview **23** is required. This interview acquires, where voluntarily provided, the user's nutritional background, sex, weight, age, ethnic background and familial health risk factors, environmental and behavioral health risk factors, medical conditions, treatments and responses, exercise, as well as subjective factors. Optionally, laboratory diagnostic information **24** may be obtained, such as blood tests for specific micronutrients and indicative of nutritional status, as well as other objective data.

Economic constraints or considerations **25** are received, e.g., a budget, for expenditures on nutritional supplements. Generally, this comprises an explicit input, but may be derived from other available information. Further, the economic constraints may be flexible, encompassing not only the nutritional supplements, but also dietary expenditures as well.

The user may also input an existing inventory **26** of nutritional supplements and optionally their acquisition cost. Otherwise, replacement cost is used to value the inventory.

A database is provided including cost information **11** for various nutritional supplements, and optionally specific information about the various nutritional supplement choices **12**, such as contents. A further option is to provide differing databases from differing vendors, allowing a user to select a vendor of choice.

At least one health model is provided which determines an optimum change in nutritional and health status **13** for the user based on acceptable changes in diet or lifestyle. Included in these changes are nutritional supplements. This model comprises a large set of formulae which represent a health status of the user, as well as models of change in health status. Each health model includes efficacy modeling for a set of nutritional supplements, as well as interaction modeling for diet, nutritional supplements, pharmaceuticals, and other factors. Thus, in this case, the health, efficacy and interaction models are unified into a single model. The user must select a health model **27** from the available choices, or may optionally hybridize existing compatible models.

Finally, a health optimization model **28** is selected which modifies the health model output based on the concept of risk and benefit. Thus, a user indicates explicitly a subjective risk tolerance, while implicit determinations of objective acceptable risk are also determined. This model is statistical in nature, and seeks to alter the aggressiveness of the proposal based on the models. It is noted that the aggressiveness weighting relies on the underlying health model. If a user seeks moderate aggressiveness in nutritional supplementation, but not necessarily high risk, then a different health model is preferably adopted which proposes the desired regimen. Generally, it would be strongly suggested to users to avoid high risk or very aggressive models except under professional supervision.

In generating the proposed nutritional supplementation **29**, it is noted that the various models may have global minima or maxima and local minima or maxima, and therefore known searching algorithms may be employed to select a preferred "operating point", i.e., to optimize the proposal. Further, it is also noted that full compliance is rarely obtained, so that the models or the health optimization model may precompensate for an expected degree of non-compliance. This expected degree of non-compliance may be estimated or based on subjective data or retrospective compliance data.

The proposal is also subject to a consistency and safety checker **30**, which seeks to prevent mistake, interaction, or abuse of the system. Thus checker operates outside of the other optimization models and independently checks a proposal for likely error or difficulty.

The system calculates an optimized proposal and outputs it to the user **31**. The user is given the opportunity to review the proposal **32**, and may alter aspects of it as desired.

The use may modify the proposal **33** with a firm constraint of a particular type, and/or a flexible "counterproposal" with respect to one or more components of the proposal. The alterations are then again processed and optimized, to yield

a new proposal. The process repeats until the proposal meets the desires of the user. However, the consistency and safety checker prevents an unsafe or unwise proposal from being generated, at least without a warning.

When accepted by the user, the final proposal is output **34**, for example printed. Upon final approval, it may be directly forwarded for order processing to a vendor **35**. Where the vendor is remote from the user, a secure electronic commerce system is employed **36**. Any data input or modified by a user is stored **37** in a personal information database **14**. While FIGS. **2** and **3** show the storage **37** occurring near the final steps of the transaction, the data may be stored at any time and preferably is at least temporarily stored as entered to prevent data loss in case of an interrupted session.

In a simplified but specific example, a consumer is a healthy 30 year old male with a balanced unsupplemented diet which meets the USDA Recommended Daily Allowances. The consumer selects an "antioxidant" health model, in which antioxidants are proposed to limit environmental toxins, limit ischemic damage due to hypoxia, and various other reputed effects. The consumer also selects a budget of $2.50 per day.

According to this model, Vitamin C, Vitamin B mixtures, Vitamin E, glutathione, as well as botanical polyphenols are considered advantageous. It is noted that glutathione and Vitamin E have caloric content, and thus, where the amounts given are significant, a reduction in normal dietary intake to compensate should be proposed.

Vitamin C is inexpensive, and often used as a filler and antioxidant in other vitamin mixtures. Thus, the vitamin C dose is maximized to subtoxic levels, generally 1500–2500 mg/day in divided doses. Vitamin B mixtures are also relatively inexpensive, and generally have a low cost at levels which avoid significant side effects. Vitamins B and C are often combined in economical dosage forms. Vitamin E is economically available, but is fat soluble, and thus the dose may be limited to about 500 IU per day. Thus, the vitamin B, C and E supplements are proposed at a reasonable "maximum" dose, leaving a significant portion of the budget remaining, e.g., about $2.25 per day.

Glutathione, while considered by many to be highly efficacious, is expensive, and thus is a cost limiting item in the optimization. Likewise, botanical polyphenols as extracts or concentrates are also considered efficacious but are costly. Therefore, one proposal might suggest that the consumer alter diet to obtain the botanical polyphenols as part of the normal diet, so that the remainder of the budgeted portion may be allocated to glutathione supplementation. Alternately, a proposal might suggest both glutathione and polyphenol nutritional supplementation in amounts proportionate to putative benefit per unit cost, to disburse the remaining budget. Thus, if glutathione is $2.50 per gram and considered to have 0.7 health benefit units per gram, and polyphenols are $20.00 per bottle of 30 100 mg capsules with 0.2 health benefit per capsule and $30.00 per bottle of 30 250 mg capsules with 0.35 health benefit units per capsule, the resulting optimization would propose one 250 mg capsule of polyphenols and 500 mg of glutathione per day. Note the drop in incremental efficacy of polyphenols according to the model and the effect of discrete dosage form availability.

In this particular case, the health model proposes an economically optimized nutritional supplementation with vitamins B, C and E as well as glutathione and polyphenols. This is, of course, a simplified example having a limited number of choices, and an actual system would have a plurality of models and a large selection of nutritional supplements available.

If the user were to seek to constrain the proposal to 10,000 mg vitamin C per day, the cost optimization might change slightly, but the consistency and safety checker would block the proposal or place a warning that such a high dose may be dangerous, e.g., renal calculi or rebound scurvy.

After the proposal is accepted, the server receives notification and payment authorization, such as from a credit card, and an order is entered with the vendor. A confirmation slip may be printed locally. The order is then processed by the vendor and shipped to the user. If a third party payor subsidizes this nutritional supplementation regimen, the order or information relating thereto may be forwarded to the payor for processing.

One month later, for example, the user may return to the kiosk **5** or other user interface system. At this time he identifies himself, and his records are retrieved. When queried about his current health status, for example, he notes objectionable skin flushing and lightheadedness after taking the water soluble vitamins. The system identifies this problem as being related to niacin flushing, and alters its proposal to a reduced flushing vitamin B (niacin) supplement formulation, for example a niacin and inositol mixture. This formulation is more expensive, and thus causes a reallocation of funds in the economic optimization. For example, less glutathione is provided.

In this case, the proposal does not identically correspond to readily available standard dosage forms of the nutritional supplements. However, a custom mixture remains an alternative. In this case, capsules containing the glutathione alone in a precise dosage, or combined with other nutrients, are custom made in sizes which correspond to the desired dose. While such custom mixture may entail a higher incremental cost than standard doses, for costly ingredients such custom mixtures may meet the requirements of the proposed nutritional supplementation better than other alternatives, and they also may provide a greater convenience utility versus ingestion of numerous pills or capsules.

Having illustrated and described the principles of the invention in a preferred embodiment, it should be apparent to those skilled in the art that the invention can be modified in arrangement and detail without departing from such principles. For example, discrete or integrated components of various types may be employed for the various parts of the apparatus, as is known to those of skill in the art. Features of the invention shown in software may also be implemented in hardware.

What I claim is:

1. A method for proposing nutritional supplementation for a person comprising the steps of:

(a) receiving health and nutritional status information relating to a person;

(b) providing information relating to a plurality of available nutritional supplements, the information comprising contents and cost;

(c) determining economic constraints for the nutritional supplementation of the person;

(d) optimizing a proposed nutritional supplementation for the person based on the health and nutritional status information, economic constraints and nutritional supplement information to improve a predicted health status of the person by nutritional supplementation with a plurality of nutritional supplements; and

(e) outputting the proposed nutritional supplementation including amounts of the plurality of nutritional supplements.

2. The method according to claim 1, further comprising the steps of determining a risk tolerance of the person and further optimizing the proposed nutritional supplementation to achieve a maximum benefit within the determined risk tolerance.

3. The method according to claim 1, further comprising the step of analyzing a proposed nutritional supplementation for health safety.

4. The method according to claim 1, further comprising the steps of receiving feedback from the person relating to the proposed nutritional supplementation and reoptimizing to generate a revised proposed nutritional supplementation.

5. The method according to claim 1, wherein the economic constraints comprise a budget.

6. The method according to claim 1, wherein the information relating to a plurality of available nutritional supplements comprises records of a stored database.

7. The method according to claim 6, wherein the database is remote from the user.

8. The method according to claim 1, further comprising the step of providing a plurality of potential optimization procedures and selecting at least one of the optimization procedures for optimizing a proposed nutritional supplementation for the person.

9. A method for proposing nutritional supplementation for a person comprising the steps of:

(a) receiving health and nutritional status information relating to a person;

(b) providing information relating to a plurality of available nutritional supplements;

(c) optimizing a proposed nutritional supplementation for the person based on the health and nutritional status information and nutritional supplement information to improve a predicted health status of the person by nutritional supplementation with a plurality of nutritional supplements;

(d) outputting the proposed nutritional supplementation including amounts of the plurality of nutritional supplements as a proposal; and

(e) transacting a sale of at least one proposed nutritional supplement with the person.

10. The method according to claim 9, further comprising the steps of receiving economic considerations from the person and optimizing the proposed nutritional supplementation further based on the economic considerations.

11. The method according to claim 10, wherein the economic considerations comprise a budget.

12. The method according to claim 9, further comprising the steps of receiving feedback from the person relating to the proposed nutritional supplementation and reoptimizing to generate a revised proposed nutritional supplementation.

13. The method according to claim 12, further comprising the step of analyzing a proposed nutritional supplementation for health safety or consistency.

14. The method according to claim 9, wherein the information relating to a plurality of available nutritional supplements, the information comprising contents and cost are stored in a database.

15. The method according to claim 9, further comprising the step of providing a plurality of potential optimization procedures and selecting at least one of the optimization procedures for optimizing a proposed nutritional supplementation for the person.

16. The method according to claim 9, wherein said sale comprises an electronic data transmission between a client system and a server system.

17. A method for proposing nutritional supplementation for a person comprising the steps of:

(a) receiving health and nutritional status information relating to a person;

(b) providing micronutrient information relating to a plurality of available nutritional supplements;

(c) providing at least one optimization procedure for optimizing a proposed nutritional supplementation for the person based on the health and nutritional status information and nutritional supplement mironutrient information to improve a predicted health status of the person by nutritional supplementation with a plurality of nutritional supplements;

(d) peforming one or more step selected from the group consisting of:

(1) selecting one of a plurality of proposed optimization procedures for optimizing a proposed nutritional supplementation for the person; and

(2) providing feedback from the person relating to the proposed nutritional supplementation and reoptimizing based on the feedback to generate a revised proposed nutritional supplementation; and

(e) outputting the proposed nutritional supplementation including amounts of the plurality of nutritional supplements as a proposal.

18. A computer readable medium, having recorded thereon a series of computer implemented instructions for controlling a computer to execute the method according to claim 17.

19. The medium according to claim 18, the method further comprising the steps of generating a graphic user interface and interacting with the person through the graphic user interface.

20. The medium according to claim 18, the method further comprising the steps of communicating between a client computer in proximity to the person and a server through a computer network.

21. The method according to claim 17, further comprising the steps of receiving economic considerations from the user and optimizing the proposed nutritional supplementation further based on the economic considerations.

22. The method according to claim 21, wherein the economic considerations comprise a budget.

23. The method according to claim 17, further comprising the steps of receiving feedback from the person relating to the proposed nutritional supplementation and reoptimizing to generate a revised proposed nutritional supplementation.

24. The method according to claim 17, wherein the information relating to a plurality of available nutritional supplements comprise a nutritional quality and a cost of a respective nutritional supplement, stored in a database.

25. The method according to claim 17, further comprising the step of providing a plurality of potential optimization procedures and selecting at least one of the optimization procedures for optimizing a proposed nutritional supplementation for the person.

26. The method according to claim 17, further comprising the step of providing nutritional supplements to the person corresponding to the proposed nutritional supplementation.

27. A method for proposing nutritional supplementation for a group of persons, comprising the steps of:

(a) receiving health and nutritional status information relating to a group of persons;

(b) providing information relating to a plurality of available nutritional supplements;

(c) optimizing a proposed single nutritional supplementation regimen for the group of persons based on the

health and nutritional status information and nutritional supplement information to improve a predicted health status of the group of persons by nutritional supplementation with a plurality of nutritional supplements; and

(d) outputting the proposed nutritional supplementation including amounts of the plurality of nutritional supplements as a proposal.

28. The method according to claim **27**, further comprising the steps of receiving economic considerations and optimizing the proposed nutritional supplementation further based on the economic considerations.

29. The method according to claim **27**, further comprising the steps of receiving feedback relating to the proposed nutritional supplementation and reoptimizing to generate a revised proposed nutritional supplementation.

30. The method according to claim **27**, wherein the information relating to a plurality of available nutritional supplements, the information comprising nutritional quality and cost are stored in a database.

31. A method for proposing nutritional supplementation for a person comprising the steps of:

(a) receiving health and nutritional status information relating to a person;

(b) providing information relating to a plurality of available nutritional supplements, said information being obtained by reference to an encoding on a container of a respective nutritional supplement;

(c) providing at least one optimization procedure for optimizing a proposed nutritional supplementation for the person based on the health and nutritional status information and nutritional supplement information to improve a predicted health status of the person by nutritional supplementation with a plurality of nutritional supplements;

(d) performing one or more step selected from the group consisting of:

(1) selecting one of a plurality of proposed optimization procedures for optimizing a proposed nutritional supplementation for the person; and

(2) providing feedback from the person relating to the proposed nutritional supplementation and reoptimizing based on the feedback to generate a revised proposed nutritional supplementation; and

(e) outputting the proposed nutritional supplementation including amounts of the plurality of nutritional supplements as a proposal.

32. The method according to claim **31**, further comprising the steps of receiving economic considerations from the person and optimizing the proposed nutritional supplementation further based on the economic considerations.

33. The method according to claim **31**, wherein the economic considerations comprise a budget.

34. The method according to claim **31**, wherein the information relating to a plurality of available nutritional supplements, the information comprising nutritional quality and cost are stored in a database.

35. The method according to claim **31**, further comprising the step of providing a plurality of potential optimization procedures and selecting at least one of the optimization procedures for optimizing a proposed nutritional supplementation for the person.

36. The method according to claim **31**, further comprising the step of, after receiving approval, selling nutritional supplements to the person corresponding to the proposed nutritional supplementation.

37. The method according to claim **36**, further comprising the step of receiving an approval of the proposed nutritional optimization through an electronic data transmission between a client system and a server system.

38. A method for proposing nutritional supplementation for a person comprising the steps of:

(a) receiving health and nutritional status information relating to a person;

(b) providing information relating to a plurality of available nutritional supplements, said information comprising at least nutritional value and cost of a respective nutritional supplement;

(c) optimizing a proposed nutritional supplementation for the person based on the health and nutritional status information and nutritional supplement information to improve a predicted health status of the person by nutritional supplementation with a plurality of nutritional supplements; and

(d) outputting the proposed nutritional supplementation including amounts of the plurality of nutritional supplements as a proposal.

39. The method according to claim **38**, further comprising the steps of receiving economic considerations from the person and optimizing the proposed nutritional supplementation further based on the economic considerations.

40. The method according to claim **39**, wherein the economic considerations comprise a budget.

41. The method according to claim **38**, further comprising the steps of receiving feedback from the person relating to the proposed nutritional supplementation and reoptimizing to generate a revised proposed nutritional supplementation.

* * * * *

APPENDIX F

(U.S. PATENT NO. 5,890,997 "ROTH")

#2844642\3

US005890997A

United States Patent [19]

Roth

[11] Patent Number: 5,890,997

[45] Date of Patent: *Apr. 6, 1999

[54] **COMPUTERIZED SYSTEM FOR THE DESIGN, EXECUTION, AND TRACKING OF EXERCISE PROGRAMS**

[76] Inventor: **Eric S. Roth**, 2445 28th St. Suite #D, Santa Monica, Calif. 90405

[*] Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

[21] Appl. No.: **802,409**

[22] Filed: **Feb. 18, 1997**

Related U.S. Application Data

[63] Continuation of Ser. No. 285,308, Aug. 3, 1994, abandoned.

[51] **Int. Cl.**[6] **A63B 71/00**

[52] **U.S. Cl.** **482/8**; 482/9; 482/902; 73/379.01; 601/23; 705/7

[58] **Field of Search** ..... 434/247; 482/1–9, 482/900–902; 73/379.01; 340/323 R; 600/520; 601/1, 23; 705/7

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,408,183	10/1983	Wills	340/323 R
4,409,992	10/1983	Sidorenko et al.	482/901
4,493,485	1/1985	Jones	73/379.01 X
4,493,488	1/1985	Panaia et al.	280/42
4,817,940	4/1989	Shaw et al.	482/9
4,828,257	5/1989	Dyer et al.	482/9 X
4,911,427	3/1990	Matsumoto et al.	482/9
5,387,164	2/1995	Brown, Jr.	482/8 X
5,410,472	4/1995	Anderson	482/9 X

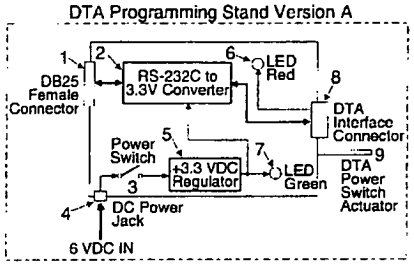
Primary Examiner—Joe H. Cheng

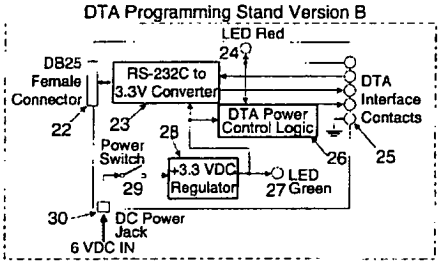
Attorney, Agent, or Firm—Matthew F. Jodziewicz

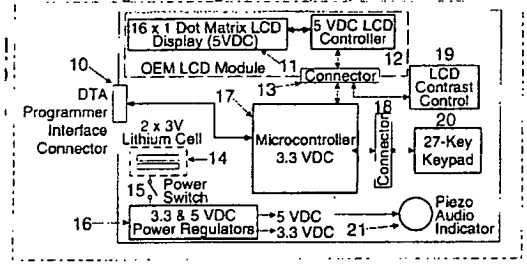
[57] **ABSTRACT**

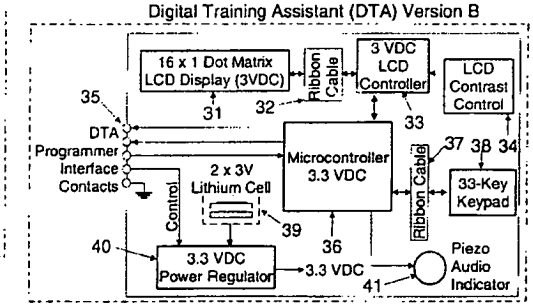
A computerized system and method for the design, execution and tracking of exercise programs including portable microprocessor controlled data controllers to instruct and record the actual computed workout for the user. A data communication link transfers data between the data controllers and a computer hosting application software and database files for the particular user, exercise and exercise regimen to create and display a customizable and comprehensive exercise system designed for the particular user.

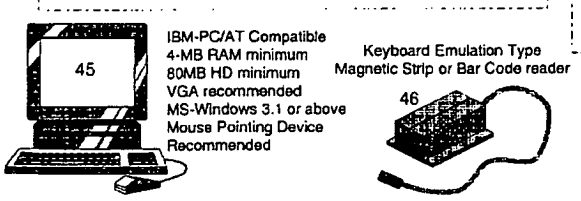
23 Claims, 35 Drawing Sheets

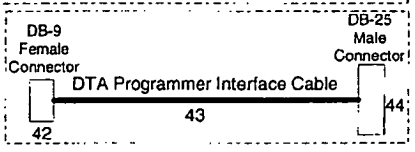












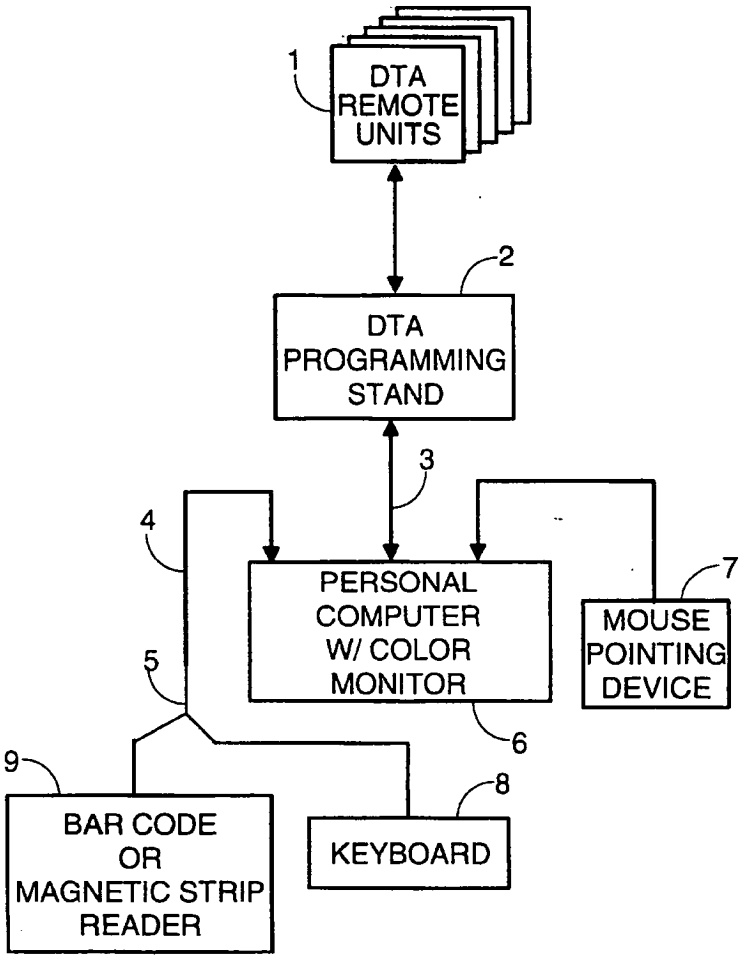

1
DTA REMOTE UNITS
2
DTA PROGRAMMING STAND
3
4
PERSONAL COMPUTER W/ COLOR MONITOR
6
7
MOUSE POINTING DEVICE
5
9
BAR CODE OR MAGNETIC STRIP READER
8
KEYBOARD

FIG. 1

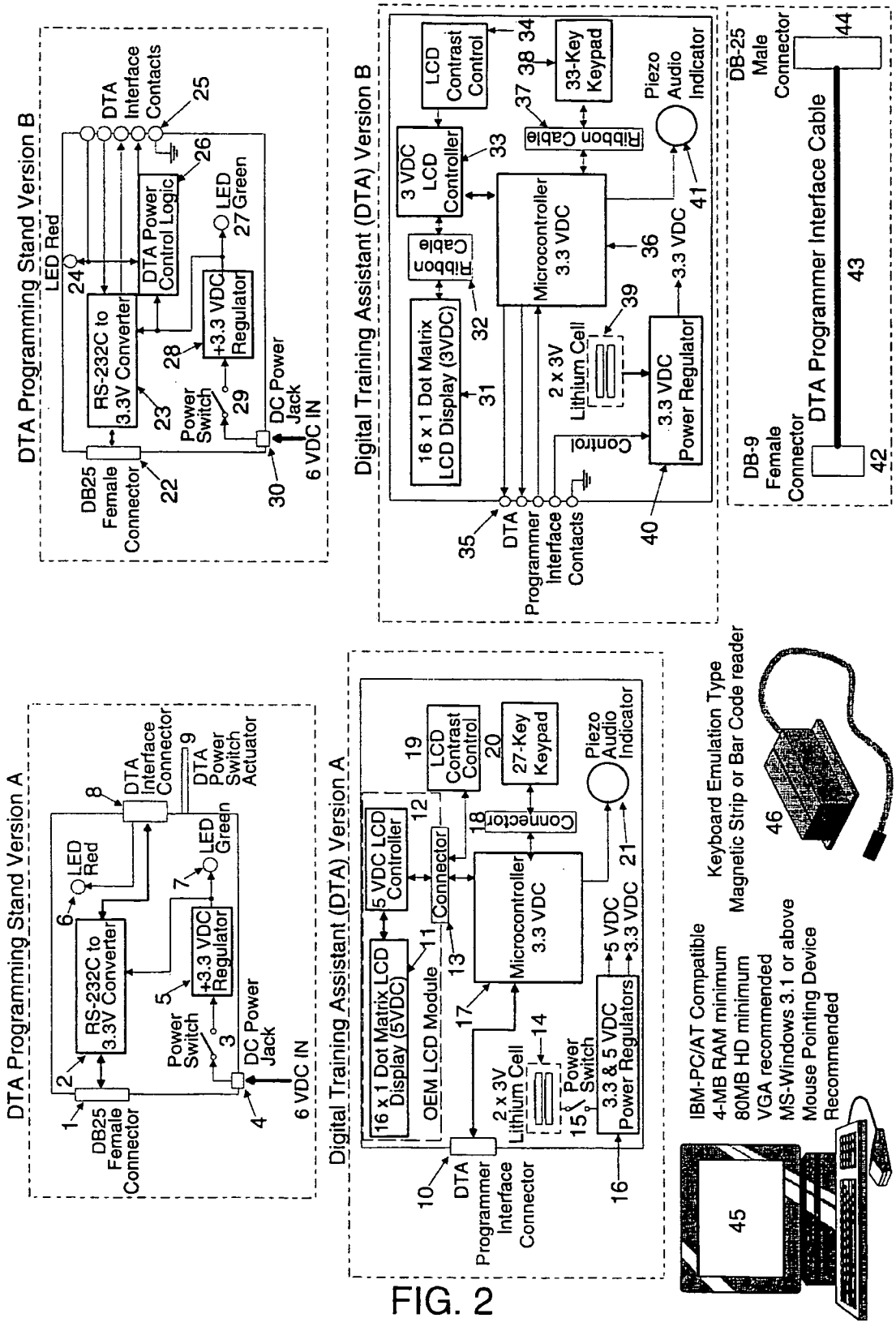

DTA Programming Stand Version B
LED Red
24
DTA Interface Contacts
25
RS-232C to 3.3V Converter
DTA Power Control Logic
26
LED 27 Green
+3.3 VDC Regulator
28
Power Switch
29
DC Power Jack
6 VDC IN
30
DB25 Female Connector
22
23
Digital Training Assistant (DTA) Version B
LCD Contrast Control
34
33-Key Keypad
37
38
Ribbon Cable
Piezo Audio Indicator
3 VDC LCD Controller
33
16 x 1 Dot Matrix LCD Display (3VDC)
31
32
Microcontroller 3.3 VDC
36
41
2 x 3V Lithium Cell
39
3.3 VDC Power Regulator
3.3 VDC
40
DTA Programmer Interface Contacts
35
DB-25 Male Connector
44
DTA Programmer Interface Cable
43
DB-9 Female Connector
42
DTA Programming Stand Version A
DTA Interface Connector
8
DTA Power Switch Actuator
9
LED Red
6
LED Green
7
RS-232C to 3.3V Converter
+3.3 VDC Regulator
5
Power Switch
3
DC Power Jack
4
6 VDC IN
2
DB25 Female Connector
1
Digital Training Assistant (DTA) Version A
LCD Contrast Control
19
27-Key Keypad
20
Piezo Audio Indicator
16 x 1 Dot Matrix LCD Display (5VDC)
OEM LCD Module
11
5 VDC LCD Controller
12
Connector
18
Microcontroller 3.3 VDC
13
17
5 VDC
3.3 VDC
21
2 x 3V Lithium Cell
14
Power Switch
15
3.3 & 5 VDC Power Regulators
16
DTA Programmer Interface Connector
10
Keyboard Emulation Type
Magnetic Strip or Bar Code reader
46
IBM-PC/AT Compatible
4-MB RAM minimum
80MB HD minimum
VGA recommended
MS-Windows 3.1 or above
Mouse Pointing Device
Recommended
45

FIG. 2

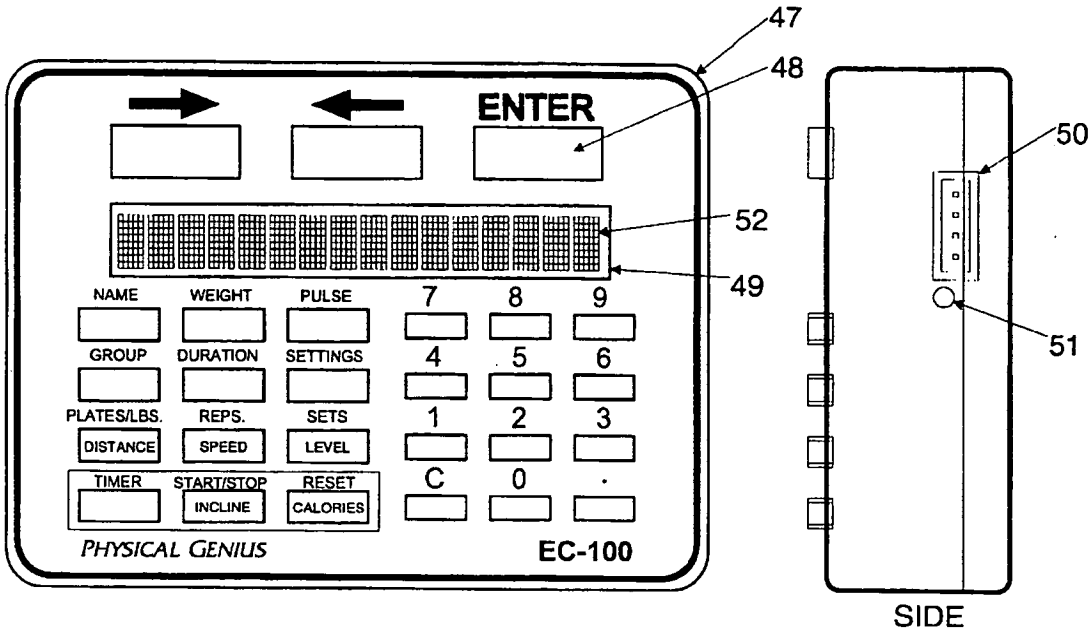

47
48
ENTER
52
49
NAME
WEIGHT
PULSE
7
8
9
GROUP
DURATION
SETTINGS
4
5
6
PLATES/LBS.
REPS.
SETS
1
2
3
DISTANCE
SPEED
LEVEL
TIMER
START/STOP
RESET
C
0
.
INCLINE
CALORIES
PHYSICAL GENIUS
EC-100
50
51
SIDE

FIG. 3

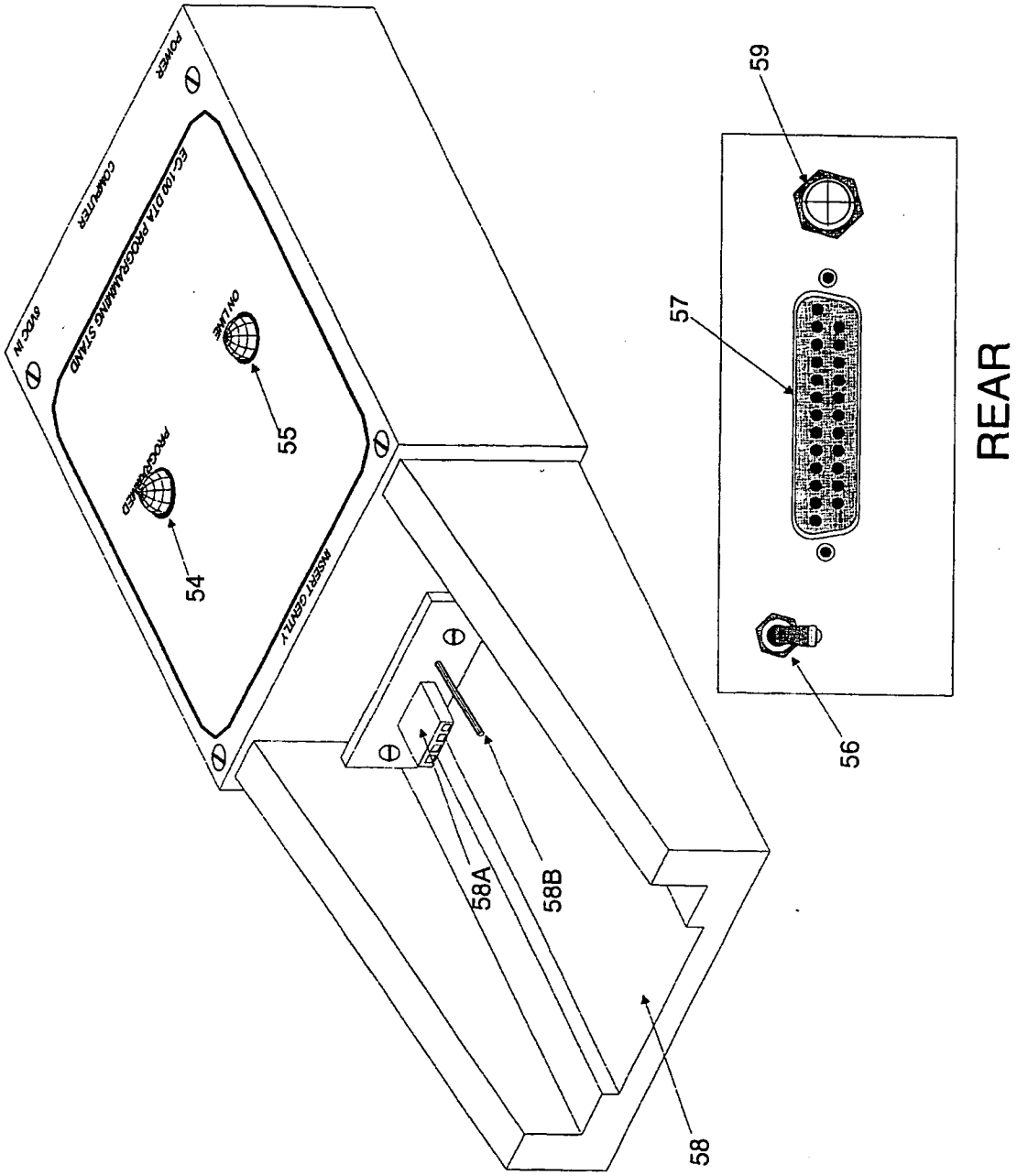


FIG. 4

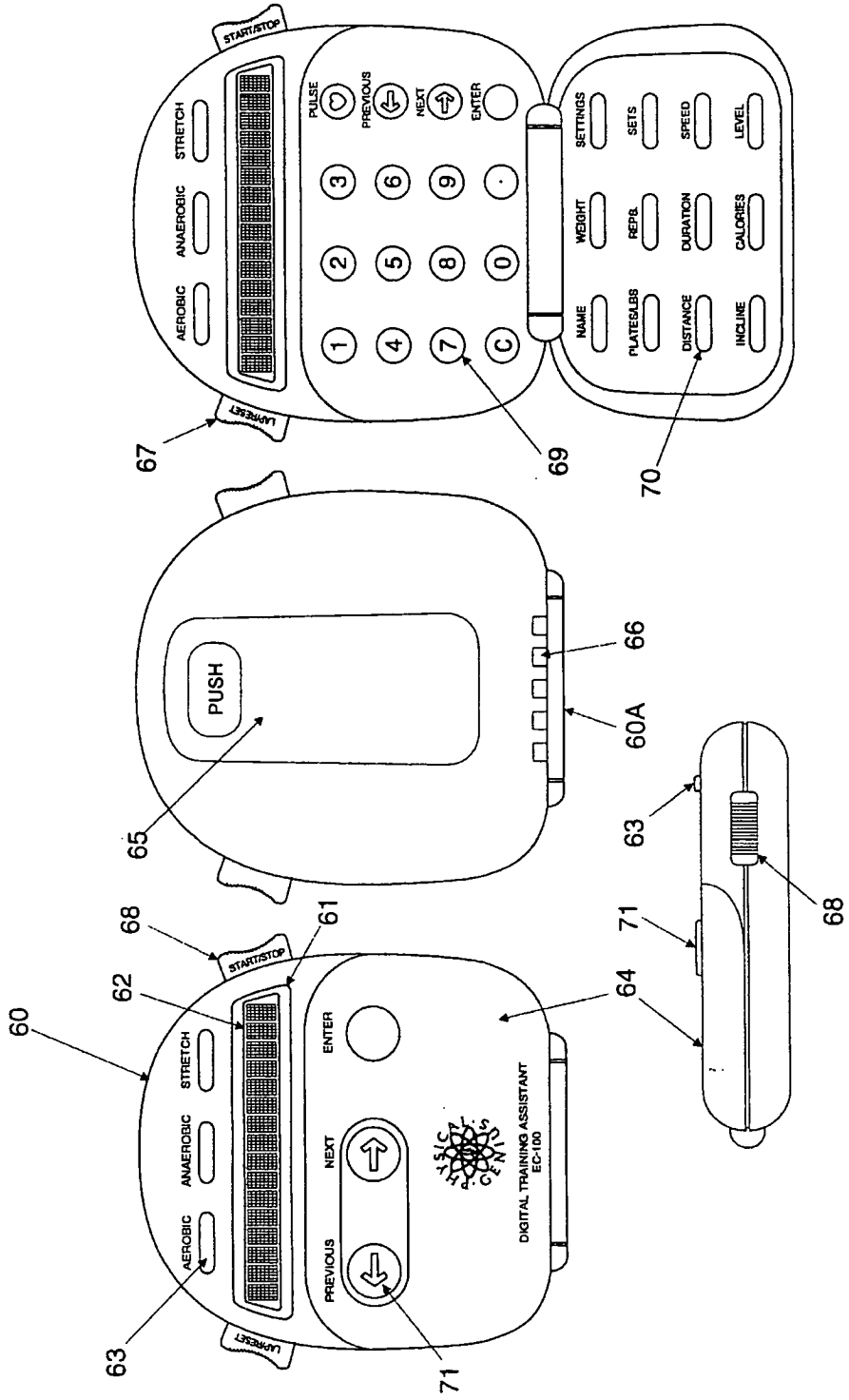

START/STOP
STRETCH
ANAEROBIC
AEROBIC
PULSE
PREVIOUS
NEXT
ENTER
SETTINGS
SETS
SPEED
LEVEL
WEIGHT
REPS
DURATION
CALORIES
NAME
PLATES/LBS
DISTANCE
INCLINE
PUSH
DIGITAL TRAINING ASSISTANT
EC-100
67
69
70
66
60A
65
63
68
71
64
61
62
60

FIG. 5

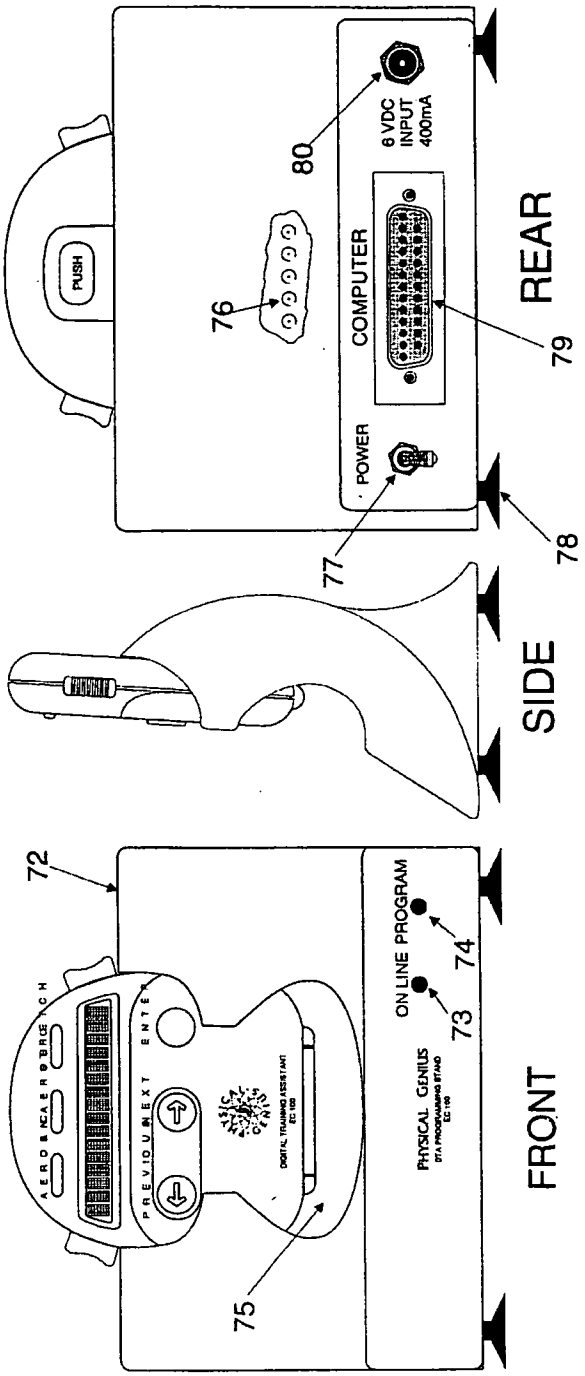

REAR
SIDE
FRONT
PUSH
COMPUTER
POWER
6 VDC INPUT 400mA
ON LINE PROGRAM
PHYSICAL GENIUS
72
73
74
75
76
77
78
79
80

FIG. 6

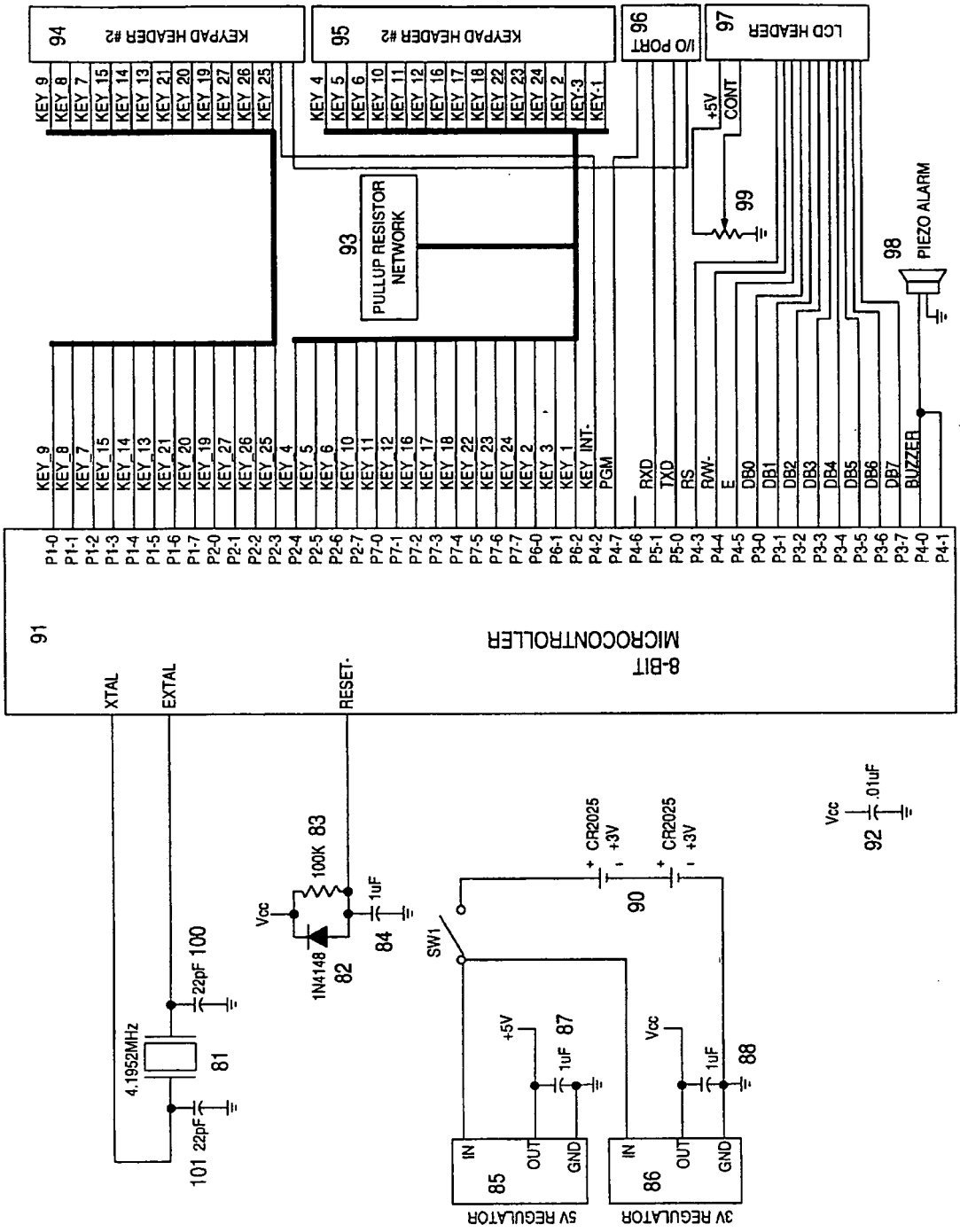

94 KEYPAD HEADER #2
95 KEYPAD HEADER #2
96 I/O PORT
97 LCD HEADER
KEY_9 KEY_8 KEY_7 KEY_15 KEY_14 KEY_13 KEY_21 KEY_20 KEY_19 KEY_27 KEY_26 KEY_25
KEY_4 KEY_5 KEY_6 KEY_10 KEY_11 KEY_12 KEY_16 KEY_17 KEY_18 KEY_22 KEY_23 KEY_24 KEY_2 KEY_3 KEY_1
+5V CONT
93 PULLUP RESISTOR NETWORK
99
98 PIEZO ALARM
KEY_INT-
PGM
RXD
TXD
RS
R/W-
E
DB0 DB1 DB2 DB3 DB4 DB5 DB6 DB7
BUZZER
P1-0 P1-1 P1-2 P1-3 P1-4 P1-5 P1-6 P1-7 P2-0 P2-1 P2-2 P2-3 P2-4 P2-5 P2-6 P2-7 P7-0 P7-1 P7-2 P7-3 P7-4 P7-5 P7-6 P7-7 P6-0 P6-1 P6-2 P4-2 P4-7 P4-6 P5-1 P5-0 P4-3 P4-4 P4-5 P3-0 P3-1 P3-2 P3-3 P3-4 P3-5 P3-6 P3-7 P4-0 P4-1
91
8-BIT MICROCONTROLLER
XTAL
EXTAL
RESET-
100K 83
1uF
Vcc
1N4148 82
84
22pF 100
4.1952MHz 81
101 22pF
CR2025 +3V
CR2025 +3V
90
SW1
+5V 1uF 87
Vcc 1uF 88
IN OUT GND 85
IN OUT GND 86
5V REGULATOR
3V REGULATOR
Vcc .01uF
92

FIG. 7

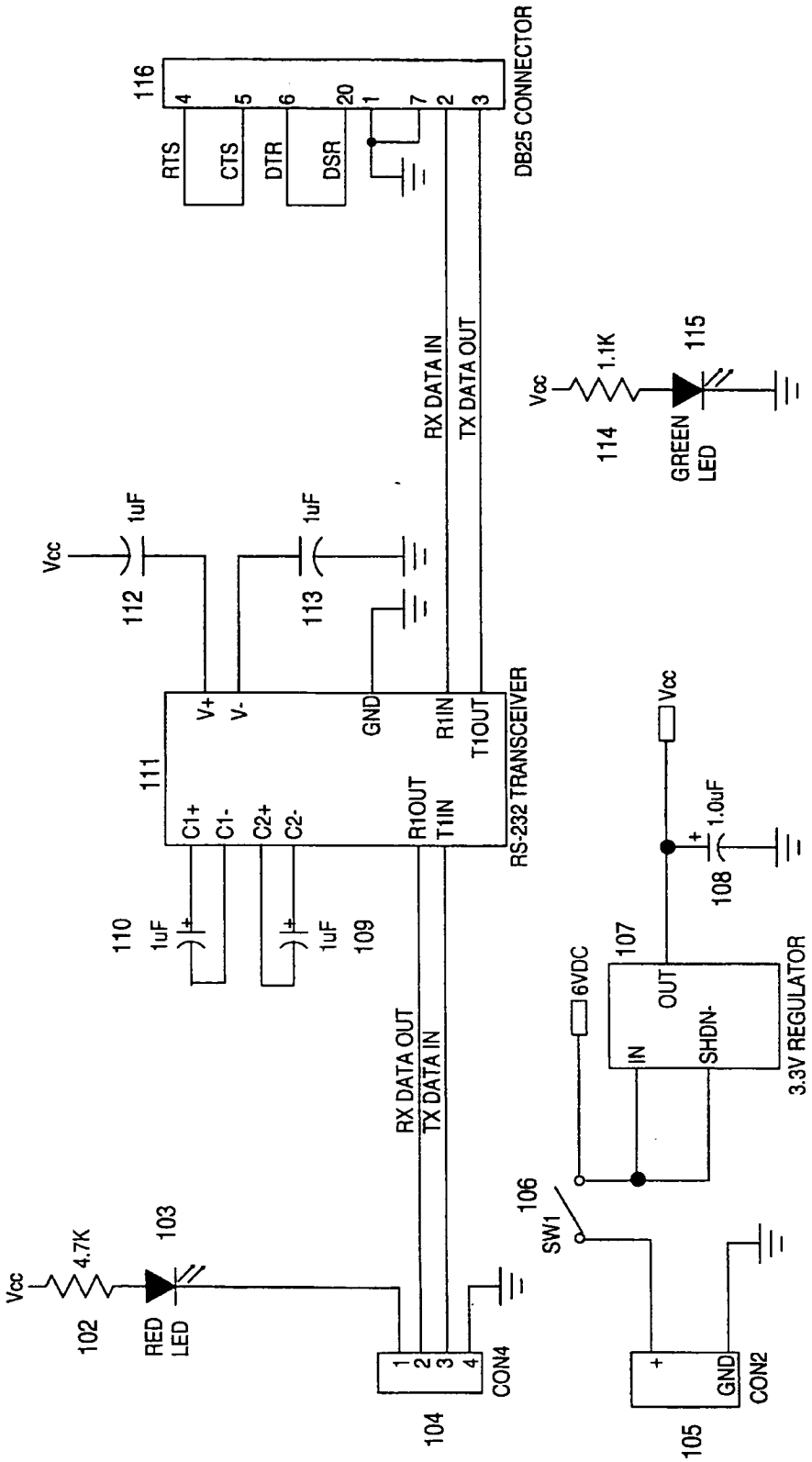

116
DB25 CONNECTOR
RTS
CTS
DTR
DSR
RX DATA IN
TX DATA OUT
Vcc
1.1K
114
GREEN LED
115
1uF
112
1uF
113
111
RS-232 TRANSCEIVER
V+
V-
GND
R1IN
T1OUT
C1+
C1-
C2+
C2-
R1OUT
T1IN
110
1uF
109
1uF
RX DATA OUT
TX DATA IN
Vcc
1.0uF
108
107
6VDC
OUT
IN
SHDN-
3.3V REGULATOR
106
SW1
Vcc
4.7K
102
103
RED LED
1
2
3
4
CON4
104
+
GND
CON2
105

FIG. 8

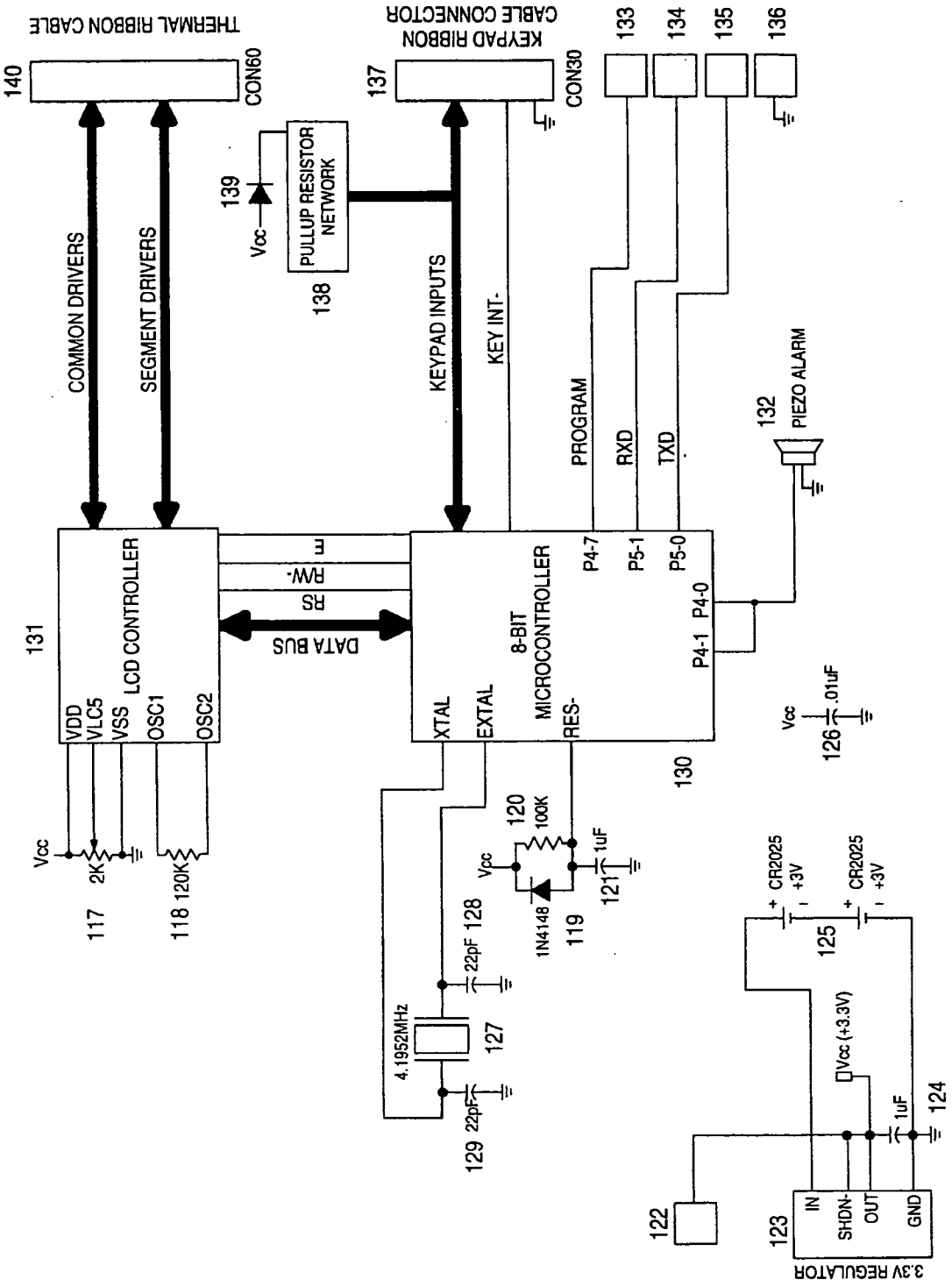

THERMAL RIBBON CABLE
KEYPAD RIBBON CABLE CONNECTOR
140
CON60
137
CON30
133
134
135
136
139
Vcc
PULLUP RESISTOR NETWORK
138
COMMON DRIVERS
SEGMENT DRIVERS
KEYPAD INPUTS
KEY INT-
PROGRAM
RXD
TXD
132
PIEZO ALARM
131
LCD CONTROLLER
VDD
VLC5
VSS
OSC1
OSC2
E
R/W-
RS
DATA BUS
8-BIT MICROCONTROLLER
XTAL
EXTAL
RES-
P4-7
P5-1
P5-0
P4-1
P4-0
130
Vcc
.01uF
126
Vcc
2K
117
120K
118
120
100K
1uF
121
1N4148
119
22pF
128
4.1952MHz
127
22pF
129
CR2025
+3V
CR2025
+3V
125
Vcc (+3.3V)
1uF
124
122
123
IN
SHDN-
OUT
GND
3.3V REGULATOR

FIG. 9

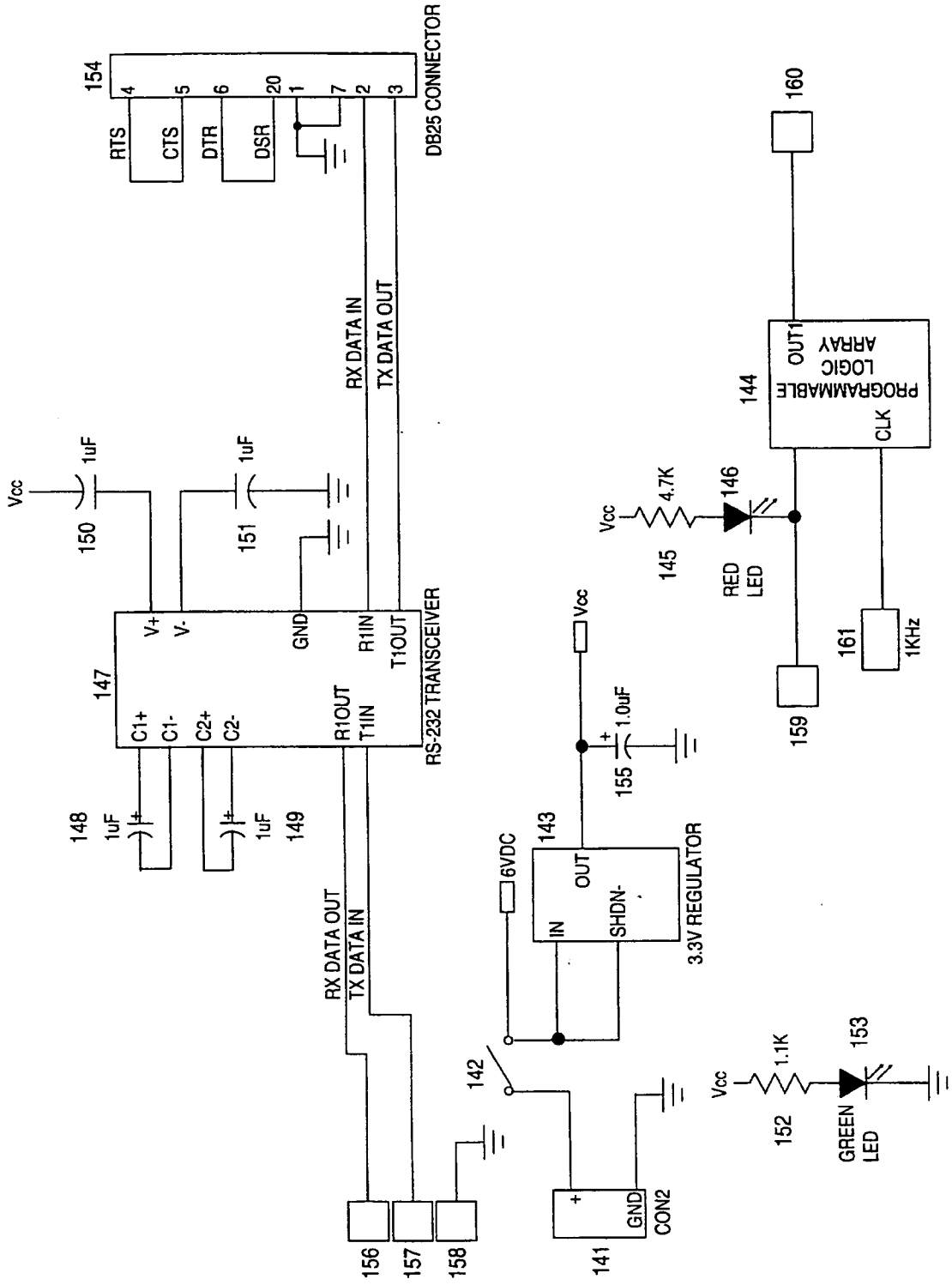

154
DB25 CONNECTOR
RTS
CTS
DTR
DSR
RX DATA IN
TX DATA OUT
Vcc
1uF
150
151
147
RS-232 TRANSCEIVER
V+
V-
GND
R1IN
T1OUT
C1+
C1-
C2+
C2-
R1OUT
T1IN
148
149
RX DATA OUT
TX DATA IN
156
157
158
142
143
6VDC
IN
OUT
SHDN-
3.3V REGULATOR
155
1.0uF
141
CON2
GND
160
144
PROGRAMMABLE LOGIC ARRAY
OUT1
CLK
146
4.7K
145
RED LED
161
1KHz
159
152
1.1K
153
GREEN LED

FIG. 10

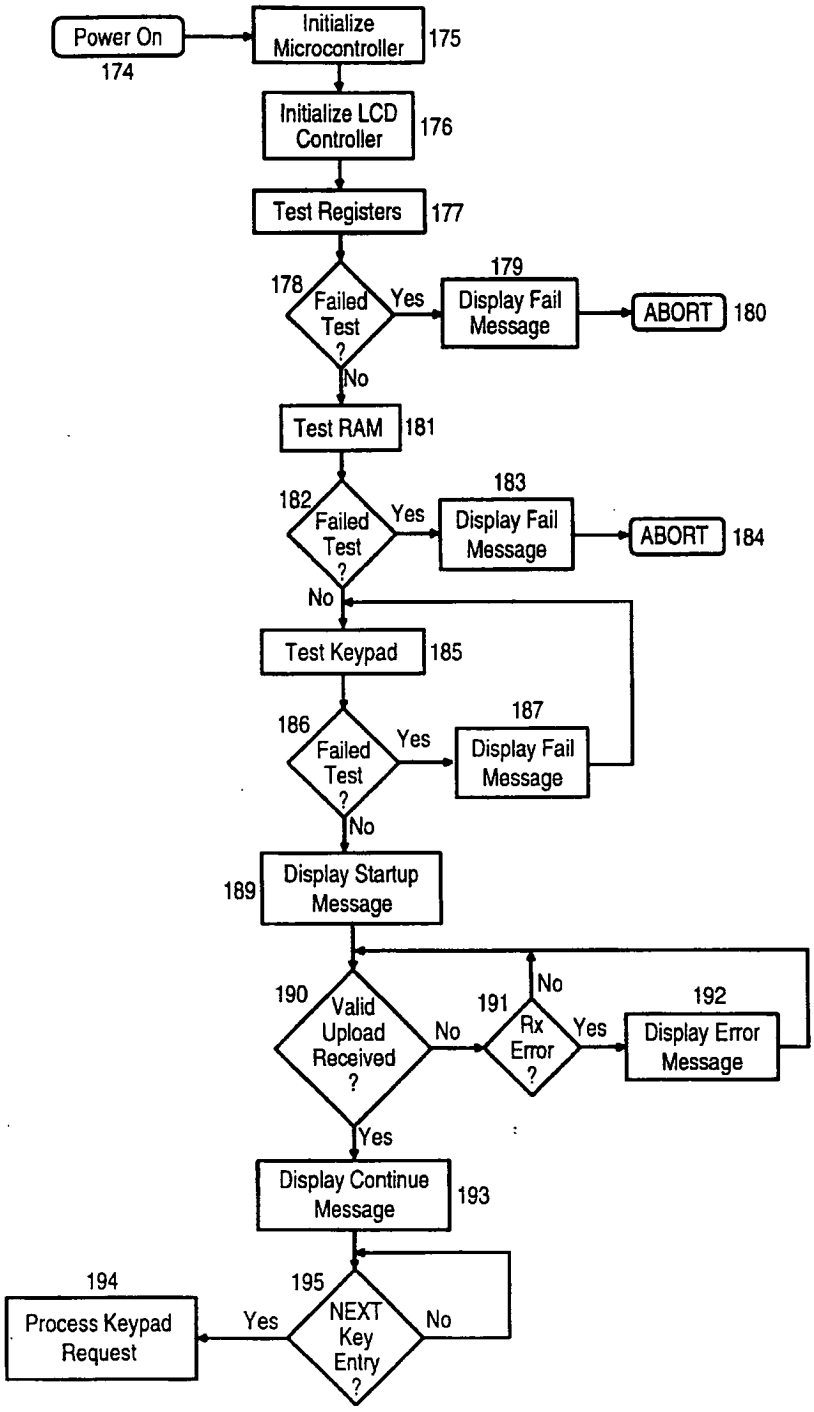

Power On
174
Initialize Microcontroller
175
Initialize LCD Controller
176
Test Registers
177
178
Failed Test ?
Yes
179
Display Fail Message
ABORT
180
No
Test RAM
181
182
Failed Test ?
Yes
183
Display Fail Message
ABORT
184
No
Test Keypad
185
186
Failed Test ?
Yes
187
Display Fail Message
No
189
Display Startup Message
190
Valid Upload Received ?
No
191
Rx Error ?
No
Yes
192
Display Error Message
Yes
Display Continue Message
193
194
Process Keypad Request
195
NEXT Key Entry ?
Yes
No

FIG. 11

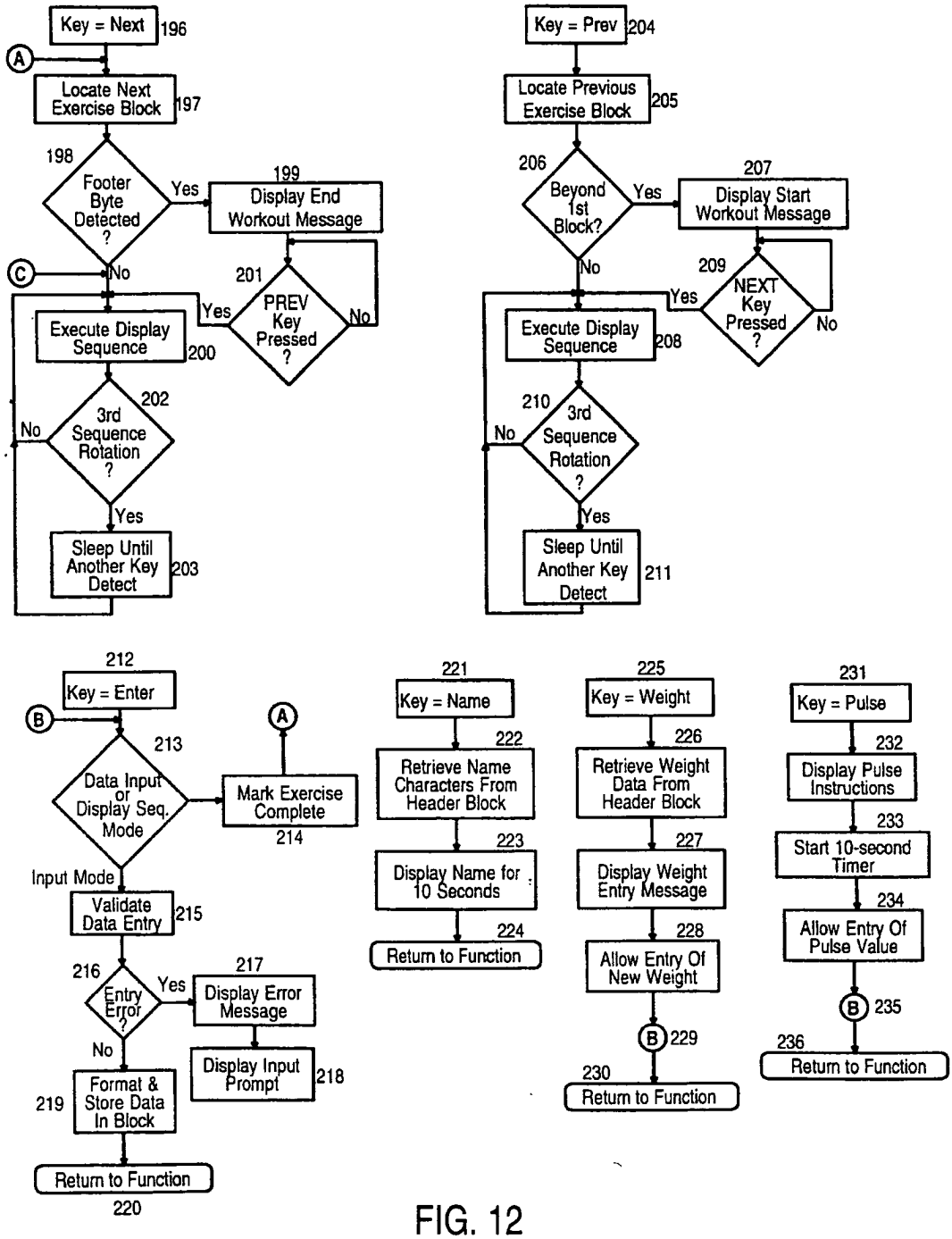

Key = Next
196
A
Locate Next Exercise Block
197
198
Footer Byte Detected ?
Yes
199
Display End Workout Message
C
No
201
PREV Key Pressed ?
Yes
No
Execute Display Sequence
200
202
3rd Sequence Rotation ?
No
Yes
Sleep Until Another Key Detect
203
Key = Prev
204
Locate Previous Exercise Block
205
206
Beyond 1st Block?
Yes
207
Display Start Workout Message
No
209
NEXT Key Pressed ?
Yes
No
Execute Display Sequence
208
210
3rd Sequence Rotation ?
No
Yes
Sleep Until Another Key Detect
211
212
Key = Enter
B
213
Data Input or Display Seq. Mode
A
Mark Exercise Complete
214
Input Mode
Validate Data Entry
215
216
Entry Error ?
Yes
217
Display Error Message
No
Display Input Prompt
218
219
Format & Store Data In Block
Return to Function
220
221
Key = Name
222
Retrieve Name Characters From Header Block
223
Display Name for 10 Seconds
224
Return to Function
225
Key = Weight
226
Retrieve Weight Data From Header Block
227
Display Weight Entry Message
228
Allow Entry Of New Weight
B
229
230
Return to Function
231
Key = Pulse
232
Display Pulse Instructions
233
Start 10-second Timer
234
Allow Entry Of Pulse Value
B
235
236
Return to Function

FIG. 12

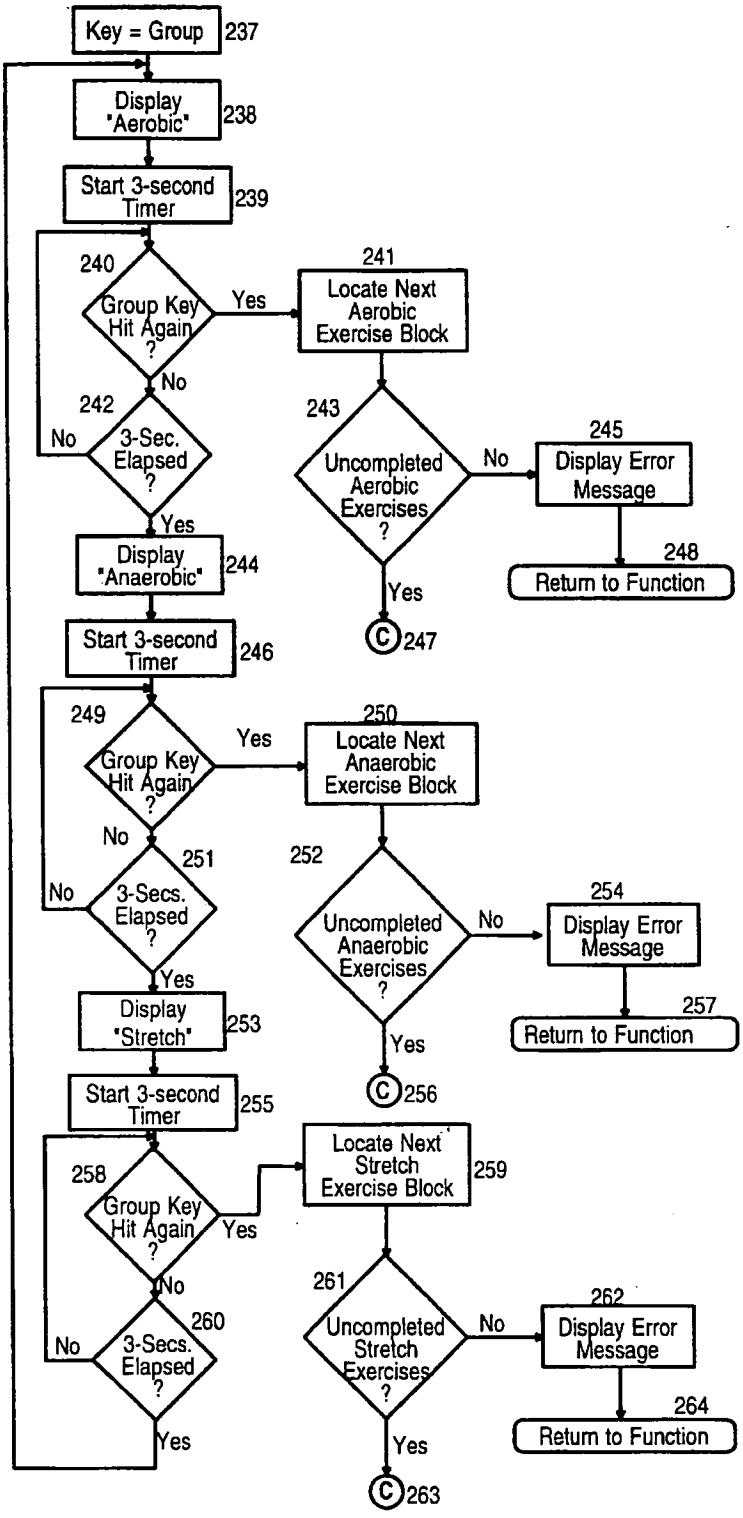

Key = Group
237
Display "Aerobic"
238
Start 3-second Timer
239
240
Group Key Hit Again ?
Yes
241
Locate Next Aerobic Exercise Block
No
242
No
3-Sec. Elapsed ?
Yes
243
Uncompleted Aerobic Exercises ?
No
245
Display Error Message
248
Return to Function
Yes
C 247
Display "Anaerobic"
244
Start 3-second Timer
246
249
Group Key Hit Again ?
Yes
250
Locate Next Anaerobic Exercise Block
No
251
No
3-Secs. Elapsed ?
Yes
252
Uncompleted Anaerobic Exercises ?
No
254
Display Error Message
257
Return to Function
Yes
C 256
Display "Stretch"
253
Start 3-second Timer
255
258
Group Key Hit Again ?
Yes
259
Locate Next Stretch Exercise Block
No
260
No
3-Secs. Elapsed ?
Yes
261
Uncompleted Stretch Exercises ?
No
262
Display Error Message
264
Return to Function
Yes
C 263

FIG. 13

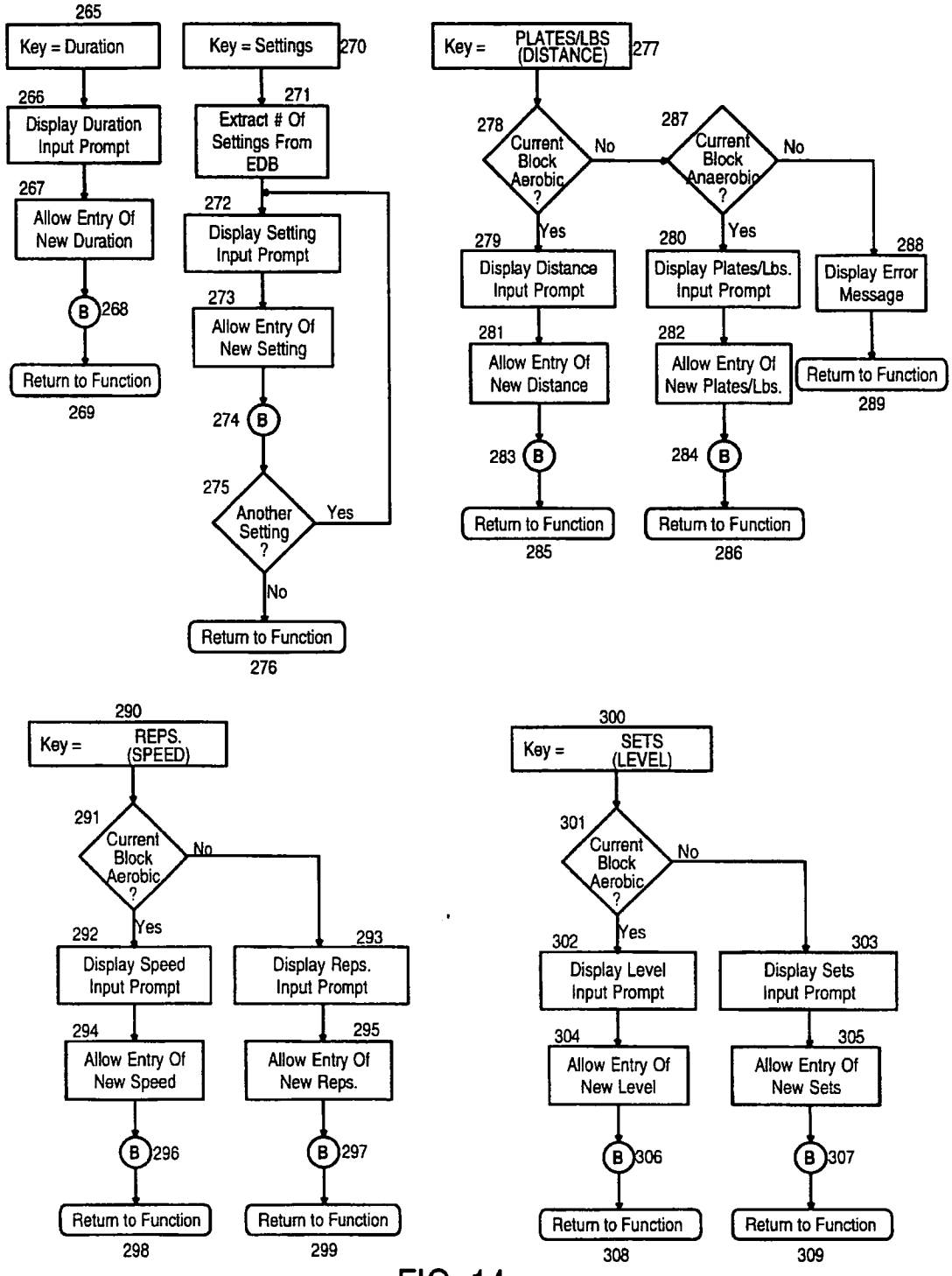

265
Key = Duration
266
Display Duration Input Prompt
267
Allow Entry Of New Duration
B 268
Return to Function
269
Key = Settings
270
271
Extract # Of Settings From EDB
272
Display Setting Input Prompt
273
Allow Entry Of New Setting
274 B
275
Another Setting ?
Yes
No
Return to Function
276
Key = PLATES/LBS (DISTANCE)
277
278
Current Block Aerobic ?
No
287
Current Block Anaerobic ?
No
Yes
279
Display Distance Input Prompt
281
Allow Entry Of New Distance
283 B
Return to Function
285
Yes
280
Display Plates/Lbs. Input Prompt
282
Allow Entry Of New Plates/Lbs.
284 B
Return to Function
286
288
Display Error Message
Return to Function
289
290
Key = REPS. (SPEED)
291
Current Block Aerobic ?
No
Yes
292
Display Speed Input Prompt
294
Allow Entry Of New Speed
B 296
Return to Function
298
293
Display Reps. Input Prompt
295
Allow Entry Of New Reps.
B 297
Return to Function
299
300
Key = SETS (LEVEL)
301
Current Block Aerobic ?
No
Yes
302
Display Level Input Prompt
304
Allow Entry Of New Level
B 306
Return to Function
308
303
Display Sets Input Prompt
305
Allow Entry Of New Sets
B 307
Return to Function
309

FIG. 14

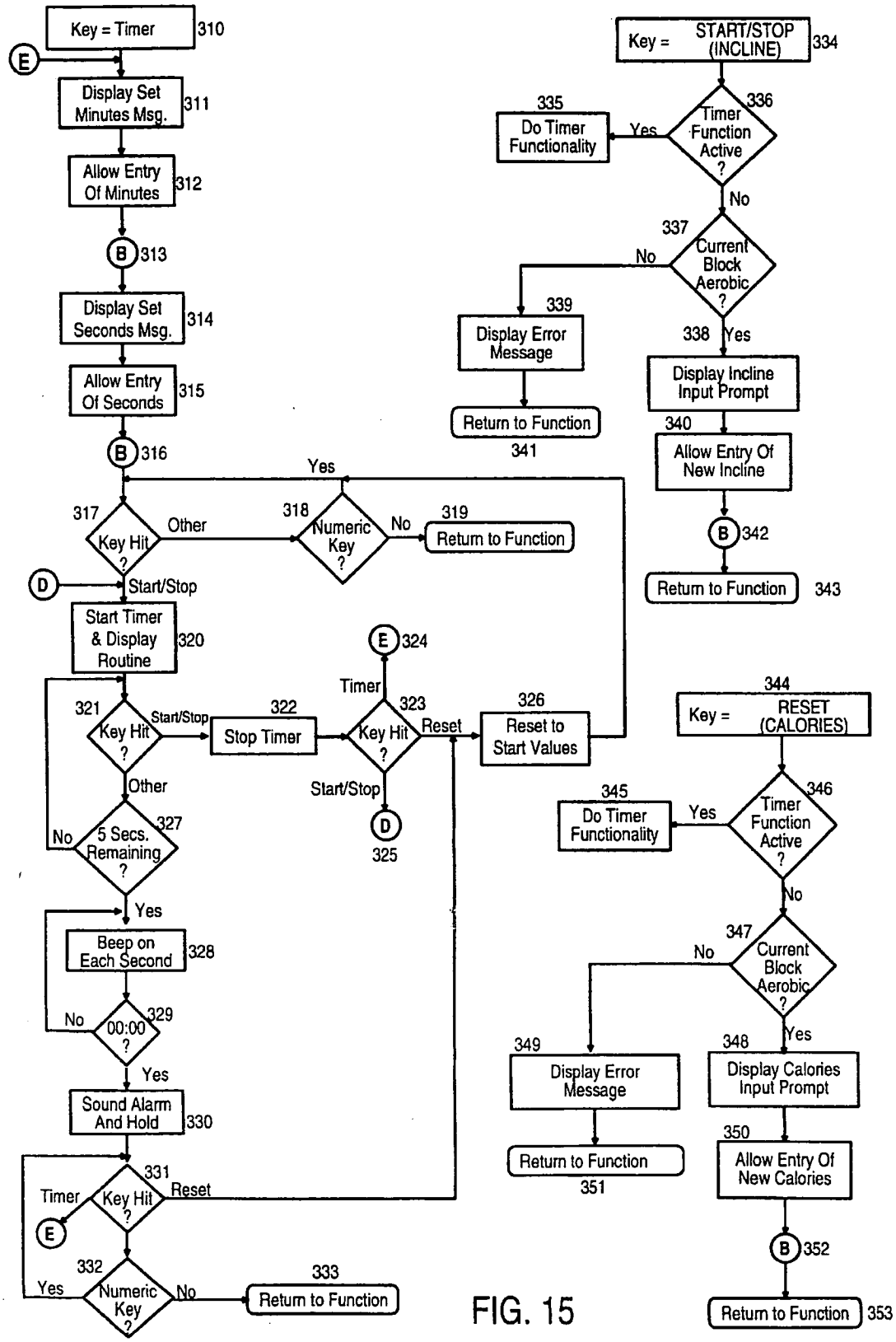

Key = Timer
310
E
Display Set Minutes Msg.
311
Allow Entry Of Minutes
312
B 313
Display Set Seconds Msg.
314
Allow Entry Of Seconds
315
B 316
317
Key Hit ?
Other
318
Numeric Key ?
No
319
Return to Function
Yes
D
Start/Stop
Start Timer & Display Routine
320
321
Key Hit ?
Start/Stop
322
Stop Timer
E 324
Timer
323
Key Hit ?
Reset
326
Reset to Start Values
Start/Stop
D 325
Other
No
327
5 Secs. Remaining ?
Yes
Beep on Each Second
328
No
329
00:00 ?
Yes
Sound Alarm And Hold
330
331
Key Hit ?
Timer
E
Reset
332
Numeric Key ?
Yes
No
333
Return to Function
Key = START/STOP (INCLINE)
334
335
Do Timer Functionality
Yes
336
Timer Function Active ?
No
337
Current Block Aerobic ?
No
339
Display Error Message
Return to Function
341
338
Yes
Display Incline Input Prompt
340
Allow Entry Of New Incline
B 342
Return to Function
343
Key = RESET (CALORIES)
344
345
Do Timer Functionality
Yes
346
Timer Function Active ?
No
347
Current Block Aerobic ?
No
349
Display Error Message
Return to Function
351
Yes
348
Display Calories Input Prompt
350
Allow Entry Of New Calories
B 352
Return to Function
353

FIG. 15

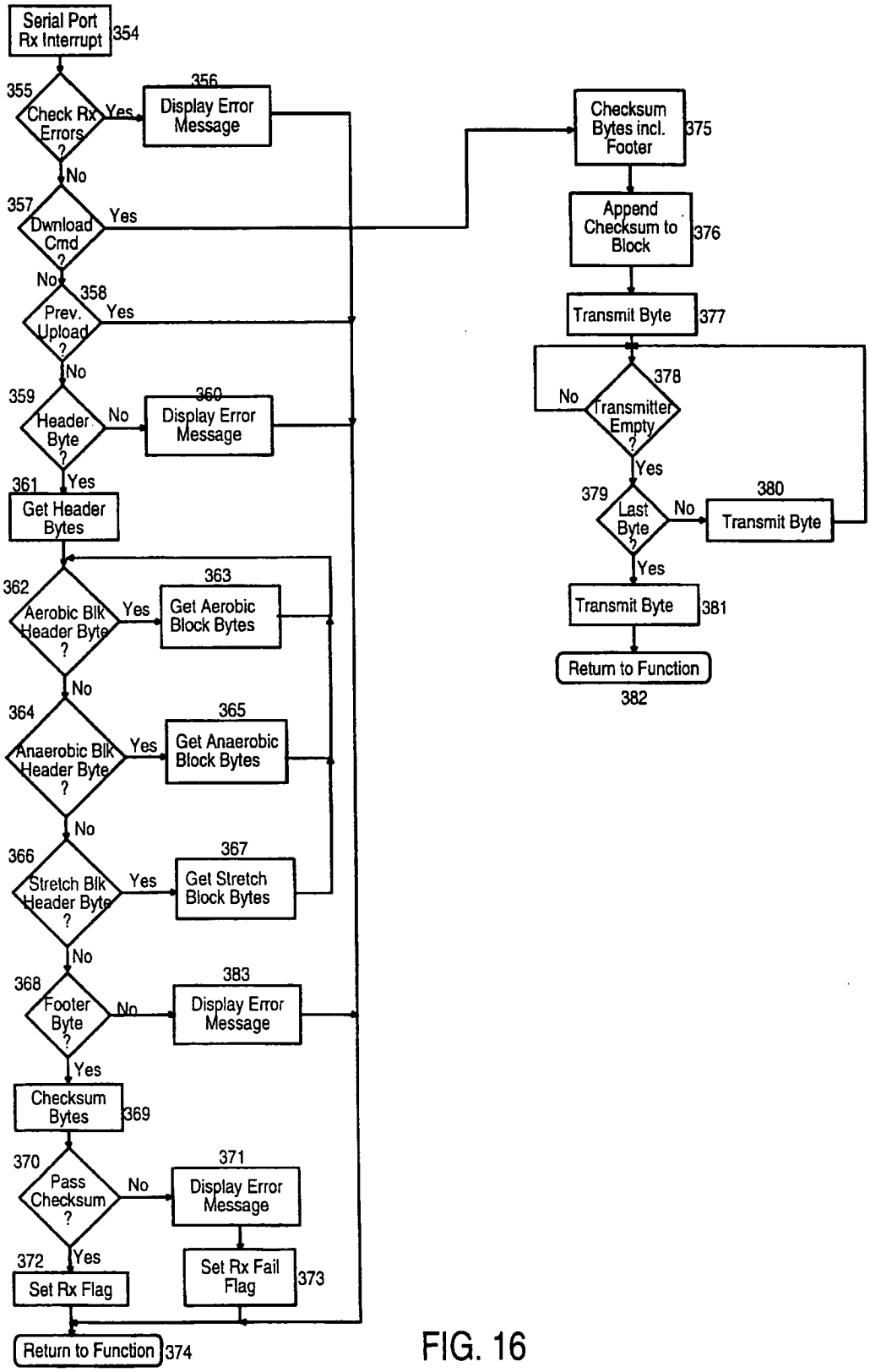

Serial Port Rx Interrupt
354
355
Check Rx Errors ?
Yes
356
Display Error Message
No
357
Dwnload Cmd ?
Yes
No
358
Prev. Upload ?
Yes
No
359
Header Byte ?
No
360
Display Error Message
Yes
361
Get Header Bytes
362
Aerobic Blk Header Byte ?
Yes
363
Get Aerobic Block Bytes
No
364
Anaerobic Blk Header Byte ?
Yes
365
Get Anaerobic Block Bytes
No
366
Stretch Blk Header Byte ?
Yes
367
Get Stretch Block Bytes
No
368
Footer Byte ?
No
383
Display Error Message
Yes
Checksum Bytes
369
370
Pass Checksum ?
No
371
Display Error Message
Yes
372
Set Rx Flag
Set Rx Fail Flag
373
Return to Function
374
Checksum Bytes incl. Footer
375
Append Checksum to Block
376
Transmit Byte
377
No
378
Transmitter Empty ?
Yes
379
Last Byte ?
No
380
Transmit Byte
Yes
Transmit Byte
381
Return to Function
382

FIG. 16

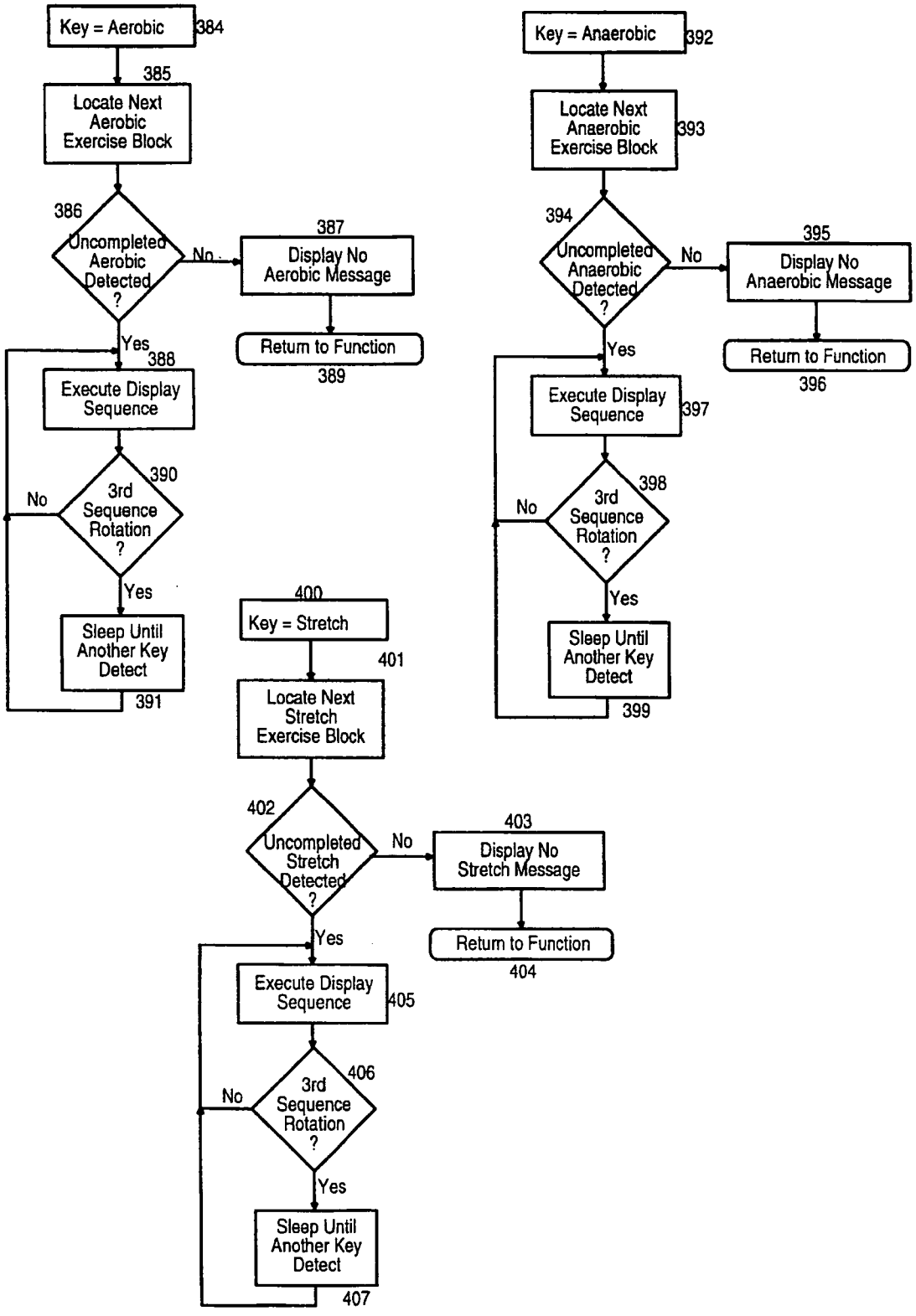

Key = Aerobic
384
385
Locate Next Aerobic Exercise Block
386
Uncompleted Aerobic Detected ?
No
387
Display No Aerobic Message
Return to Function
389
Yes
388
Execute Display Sequence
390
3rd Sequence Rotation ?
No
Yes
Sleep Until Another Key Detect
391
Key = Anaerobic
392
Locate Next Anaerobic Exercise Block
393
394
Uncompleted Anaerobic Detected ?
No
395
Display No Anaerobic Message
Return to Function
396
Yes
Execute Display Sequence
397
398
3rd Sequence Rotation ?
No
Yes
Sleep Until Another Key Detect
399
400
Key = Stretch
401
Locate Next Stretch Exercise Block
402
Uncompleted Stretch Detected ?
No
403
Display No Stretch Message
Return to Function
404
Yes
Execute Display Sequence
405
406
3rd Sequence Rotation ?
No
Yes
Sleep Until Another Key Detect
407

FIG. 17

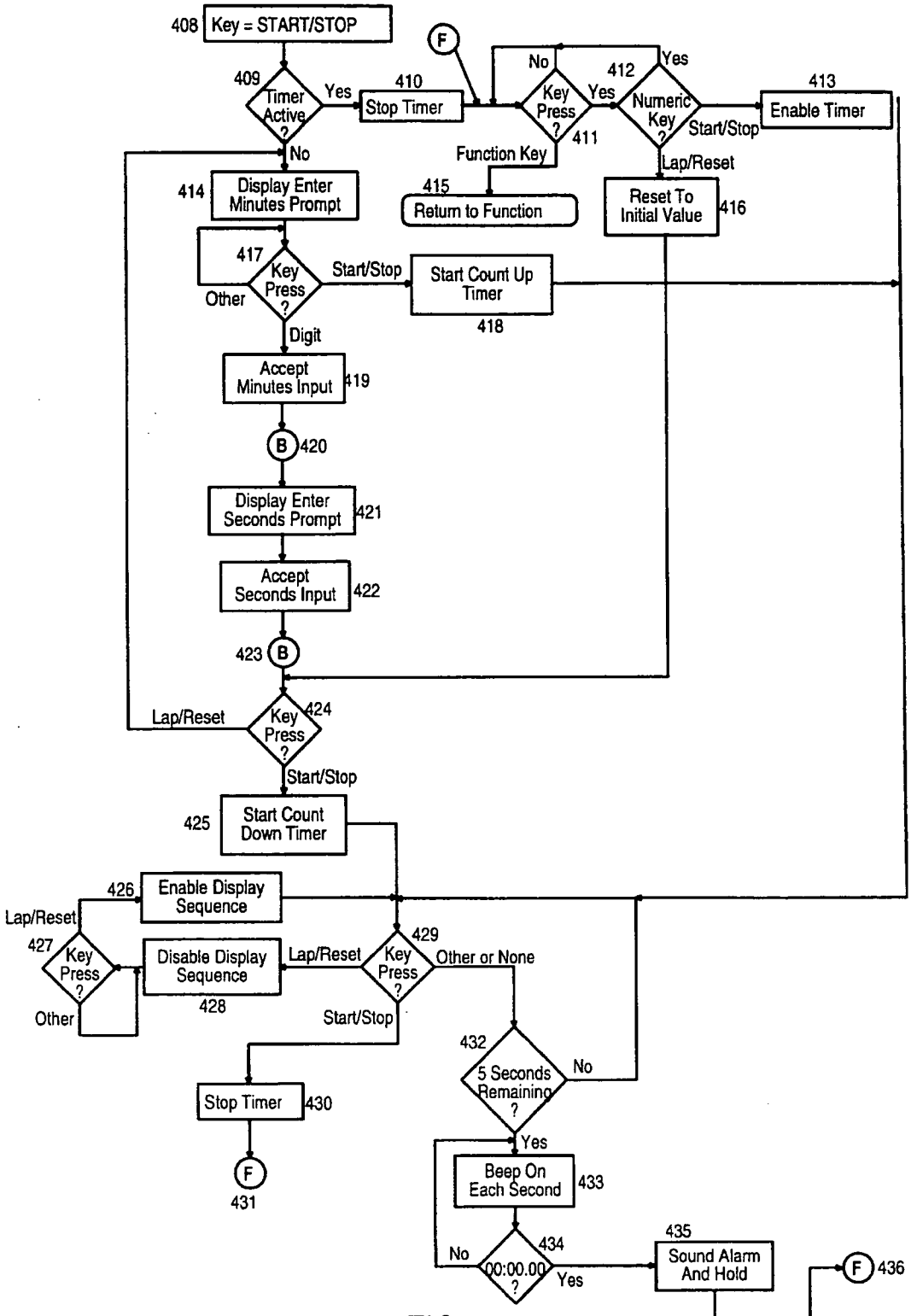

408 Key = START/STOP
409 Timer Active ?
Yes
410 Stop Timer
No
F
No
Key Press ?
411
Yes
412 Numeric Key ?
Yes
Start/Stop
413 Enable Timer
Lap/Reset
Function Key
415 Return to Function
Reset To Initial Value
416
414 Display Enter Minutes Prompt
417 Key Press ?
Other
Start/Stop
Start Count Up Timer
418
Digit
Accept Minutes Input
419
B 420
Display Enter Seconds Prompt
421
Accept Seconds Input
422
423 B
Lap/Reset
Key Press ?
424
Start/Stop
425 Start Count Down Timer
426 Enable Display Sequence
Lap/Reset
427 Key Press ?
Other
Disable Display Sequence
428
Lap/Reset
429 Key Press ?
Other or None
Start/Stop
Stop Timer
430
F
431
432 5 Seconds Remaining ?
No
Yes
Beep On Each Second
433
No
434 00:00.00 ?
Yes
435 Sound Alarm And Hold
F 436

FIG. 18

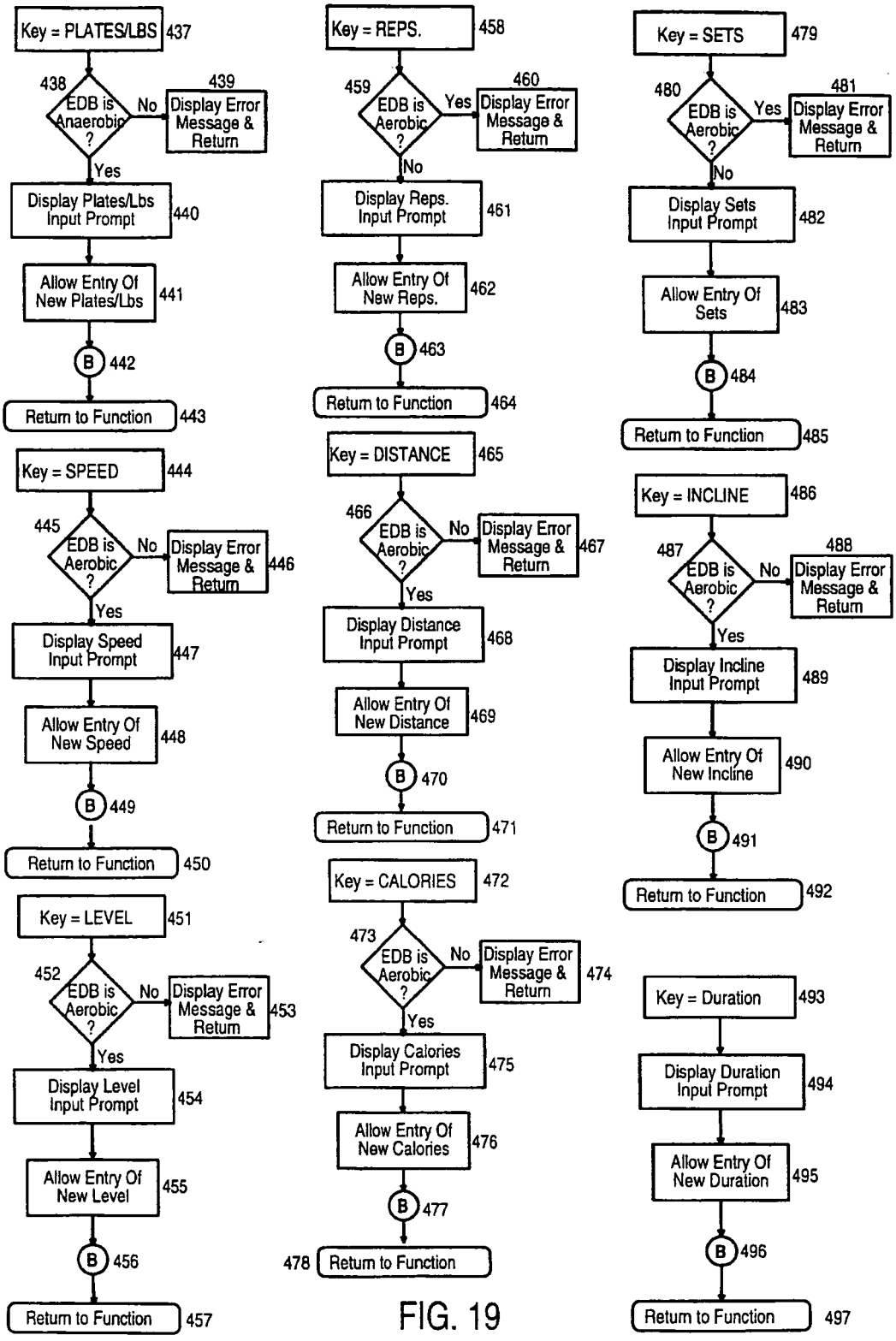

Key = PLATES/LBS
437
438
EDB is Anaerobic ?
No
439
Display Error Message & Return
Yes
Display Plates/Lbs Input Prompt
440
Allow Entry Of New Plates/Lbs
441
B
442
Return to Function
443
Key = SPEED
444
445
EDB is Aerobic ?
No
Display Error Message & Return
446
Yes
Display Speed Input Prompt
447
Allow Entry Of New Speed
448
B
449
Return to Function
450
Key = LEVEL
451
452
EDB is Aerobic ?
No
Display Error Message & Return
453
Yes
Display Level Input Prompt
454
Allow Entry Of New Level
455
B
456
Return to Function
457
Key = REPS.
458
459
EDB is Aerobic ?
Yes
460
Display Error Message & Return
No
Display Reps. Input Prompt
461
Allow Entry Of New Reps.
462
B
463
Return to Function
464
Key = DISTANCE
465
466
EDB is Aerobic ?
No
Display Error Message & Return
467
Yes
Display Distance Input Prompt
468
Allow Entry Of New Distance
469
B
470
Return to Function
471
Key = CALORIES
472
473
EDB is Aerobic ?
No
Display Error Message & Return
474
Yes
Display Calories Input Prompt
475
Allow Entry Of New Calories
476
B
477
478
Return to Function
Key = SETS
479
480
EDB is Aerobic ?
Yes
481
Display Error Message & Return
No
Display Sets Input Prompt
482
Allow Entry Of Sets
483
B
484
Return to Function
485
Key = INCLINE
486
487
EDB is Aerobic ?
No
488
Display Error Message & Return
Yes
Display Incline Input Prompt
489
Allow Entry Of New Incline
490
B
491
Return to Function
492
Key = Duration
493
Display Duration Input Prompt
494
Allow Entry Of New Duration
495
B
496
Return to Function
497

FIG. 19

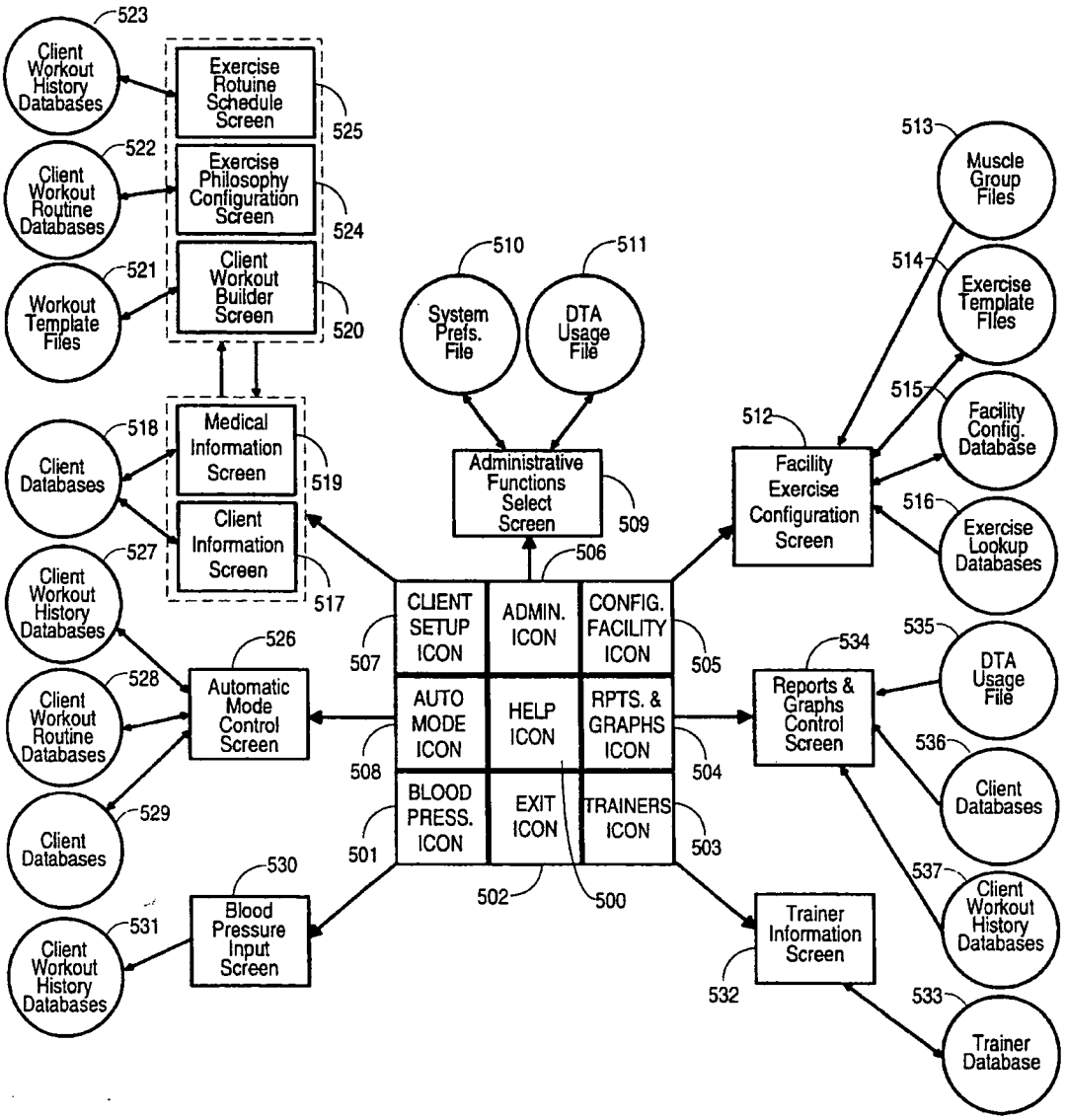

523
Client Workout History Databases
Exercise Rotuine Schedule Screen
525
522
Client Workout Routine Databases
Exercise Philosophy Configuration Screen
524
521
Workout Template Files
Client Workout Builder Screen
520
510
System Prefs. File
511
DTA Usage File
513
Muscle Group Files
514
Exercise Template Files
515
Facility Config. Database
516
Exercise Lookup Databases
512
Facility Exercise Configuration Screen
518
Client Databases
Medical Information Screen
519
Client Information Screen
517
Administrative Functions Select Screen
509
506
527
Client Workout History Databases
526
Automatic Mode Control Screen
528
Client Workout Routine Databases
529
Client Databases
530
Blood Pressure Input Screen
531
Client Workout History Databases
CLIENT SETUP ICON
507
ADMIN. ICON
CONFIG. FACILITY ICON
505
AUTO MODE ICON
508
HELP ICON
RPTS. & GRAPHS ICON
504
BLOOD PRESS. ICON
501
EXIT ICON
502
500
TRAINERS ICON
503
534
Reports & Graphs Control Screen
535
DTA Usage File
536
Client Databases
537
Client Workout History Databases
Trainer Information Screen
532
533
Trainer Database

FIG. 20

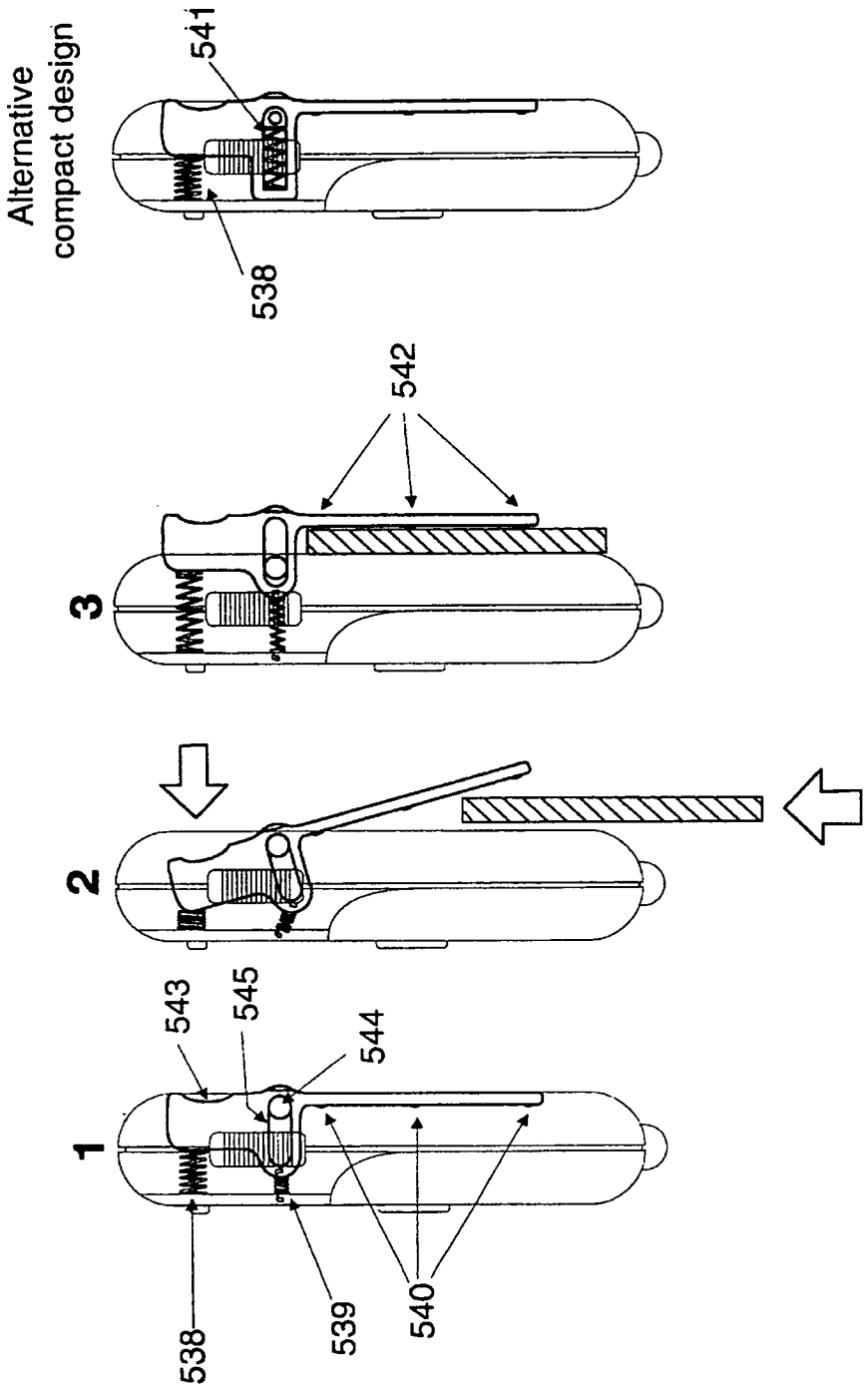
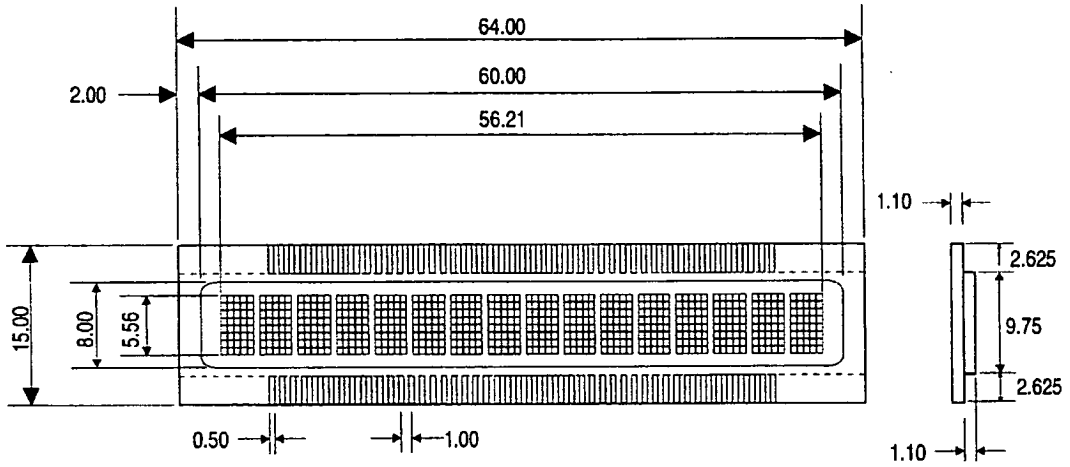
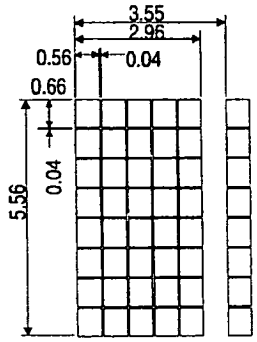

Alternative compact design
541
538
542
3
2
543
545
544
1
538
539
540

FIG. 21





DOT MATRIX DETAIL
SCALE 8:1

FIG. 22

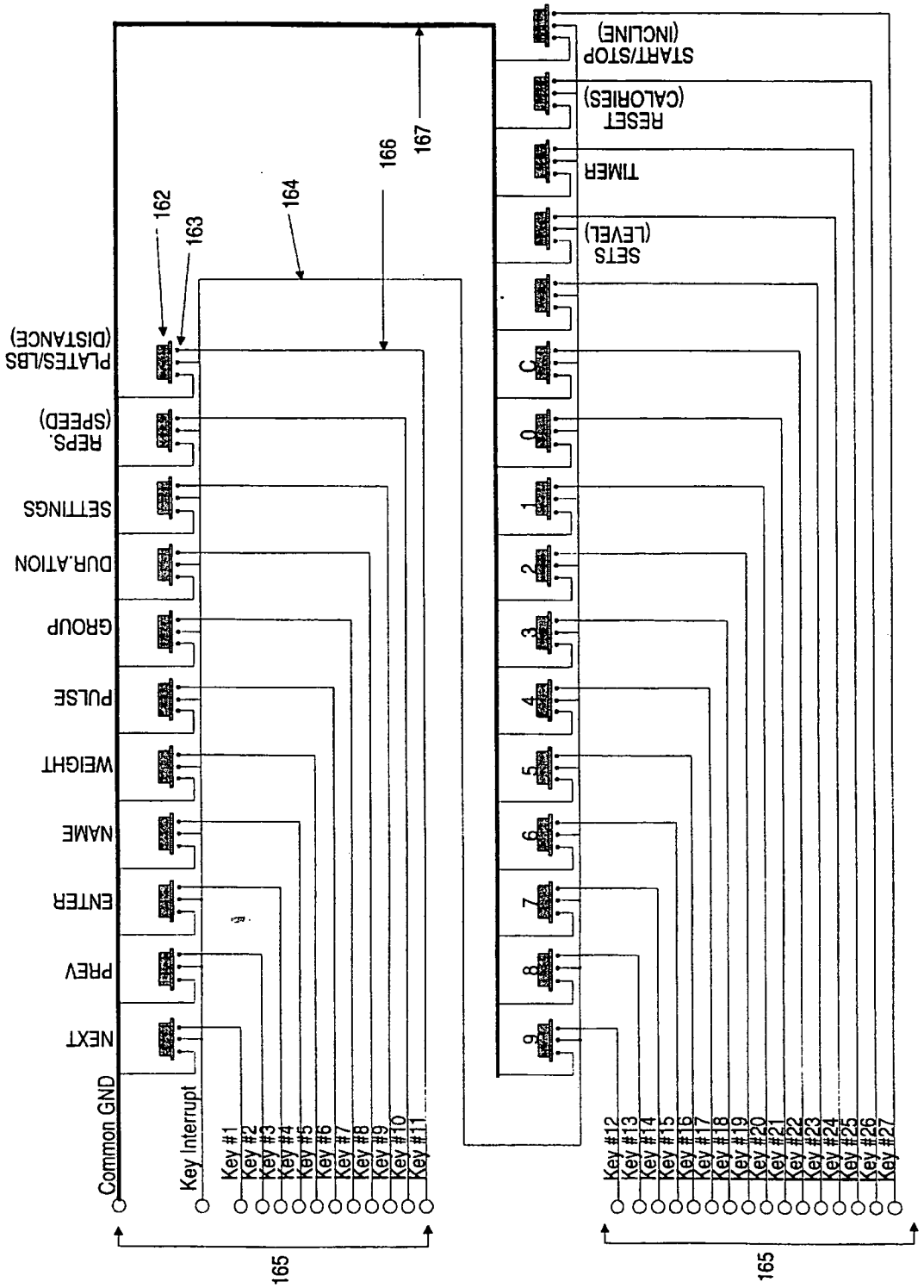

162
163
164
166
167
PLATES/LBS (DISTANCE)
REPS. (SPEED)
SETTINGS
DUR.ATION
GROUP
PULSE
WEIGHT
NAME
ENTER
PREV
NEXT
Common GND
Key Interrupt
Key #1
Key #2
Key #3
Key #4
Key #5
Key #6
Key #7
Key #8
Key #9
Key #10
Key #11
START/STOP (INCLINE)
RESET (CALORIES)
TIMER
SETS (LEVEL)
0
1
2
3
4
5
6
7
8
9
Key #12
Key #13
Key #14
Key #15
Key #16
Key #17
Key #18
Key #19
Key #20
Key #21
Key #22
Key #23
Key #24
Key #25
Key #26
Key #27
165
165

FIG. 23

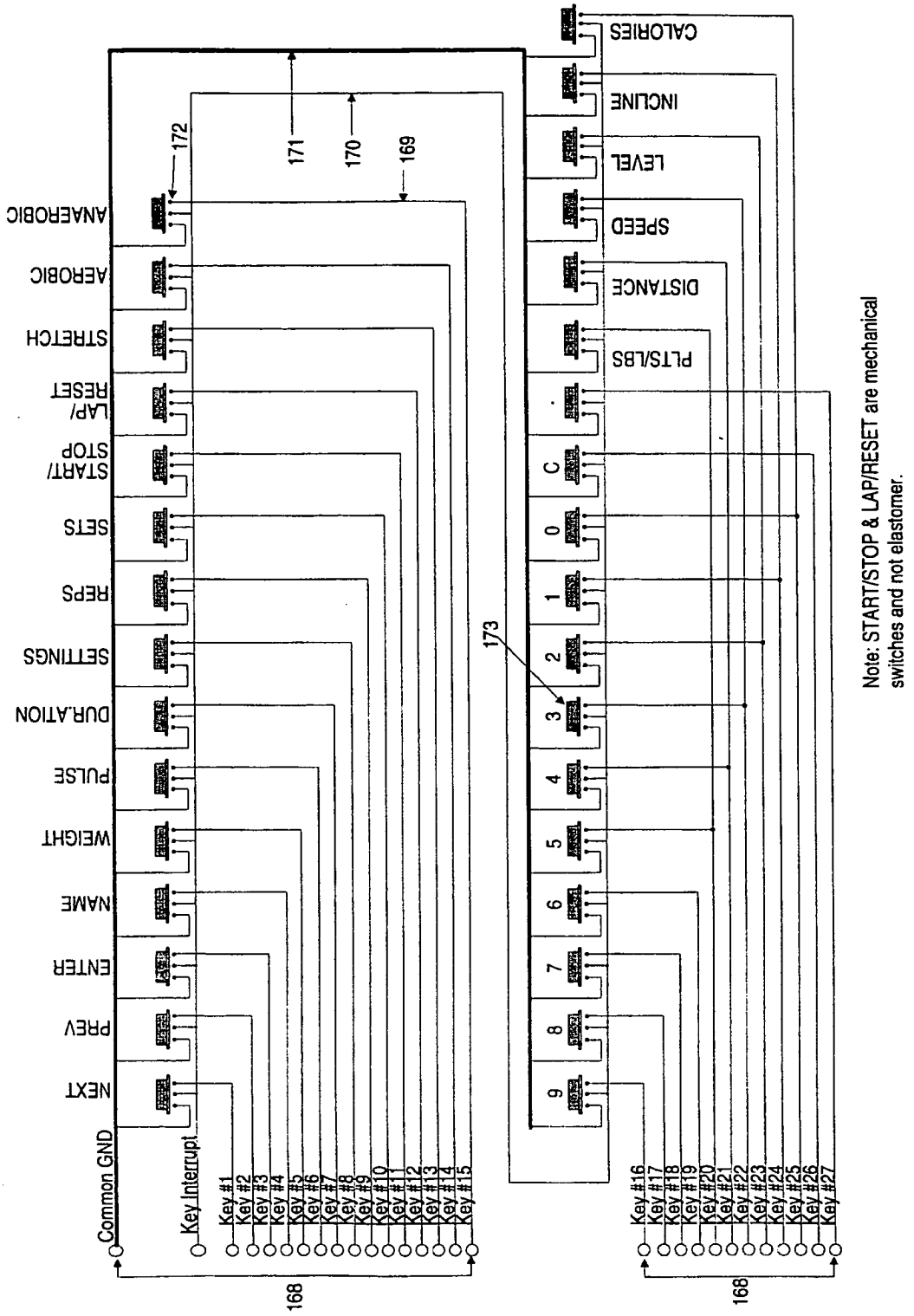

CALORIES
INCLINE
LEVEL
SPEED
DISTANCE
PLTS/LBS
C
0
1
2
3
4
5
6
7
8
9
ANAEROBIC
AEROBIC
STRETCH
LAP/ RESET
START/ STOP
SETS
REPS
SETTINGS
DUR.ATION
PULSE
WEIGHT
NAME
ENTER
PREV
NEXT
172
171
170
169
173
Common GND
Key Interrupt
Key #1
Key #2
Key #3
Key #4
Key #5
Key #6
Key #7
Key #8
Key #9
Key #10
Key #11
Key #12
Key #13
Key #14
Key #15
Key #16
Key #17
Key #18
Key #19
Key #20
Key #21
Key #22
Key #23
Key #24
Key #25
Key #26
Key #27
168
168
Note: START/STOP & LAP/RESET are mechanical switches and not elastomer.

FIG. 24

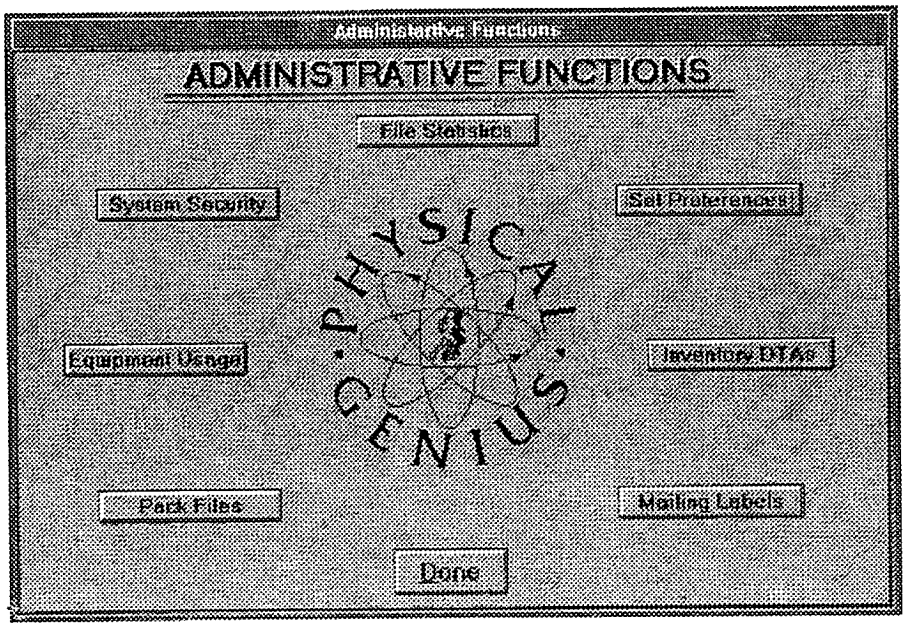

Administrative Functions
ADMINISTRATIVE FUNCTIONS
File Statistics
System Security
Set Preferences
PHYSICAL GENIUS
Equipment Usage
Inventory DTAs
Park Files
Mailing Labels
Done

FIG. 25

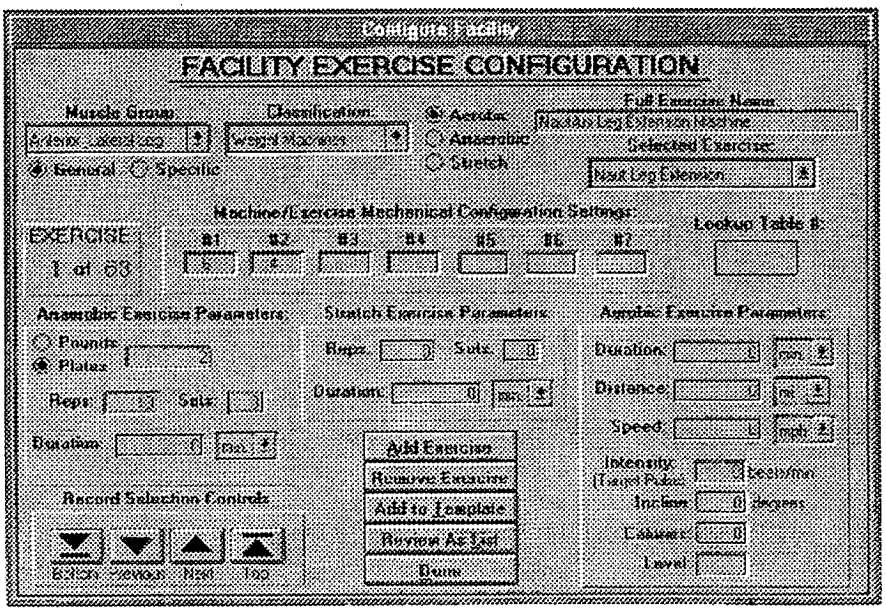

FACILITY EXERCISE CONFIGURATION
Muscle Group
Classification
Acrobic
Anaerobic
Stretch
General
Specific
Full Exercise Name
Selected Exercise:
Machine/Exercise Mechanical Configuration Settings:
Lookup Table #
EXERCISE
1 of 69
#1
#2
#3
#4
#5
#6
#7
Anaerobic Exercise Parameters:
Pounds
Plates
Reps:
Sets:
Duration:
Stretch Exercise Parameters:
Reps:
Sets:
Duration:
Aerobic Exercise Parameters:
Duration:
Distance:
Speed:
Intensity
Incline:
Calories:
Level:
Add Exercise
Remove Exercise
Add to Template
Review As Set
Done
Record Selection Controls:
Prev
Next

FIG. 26

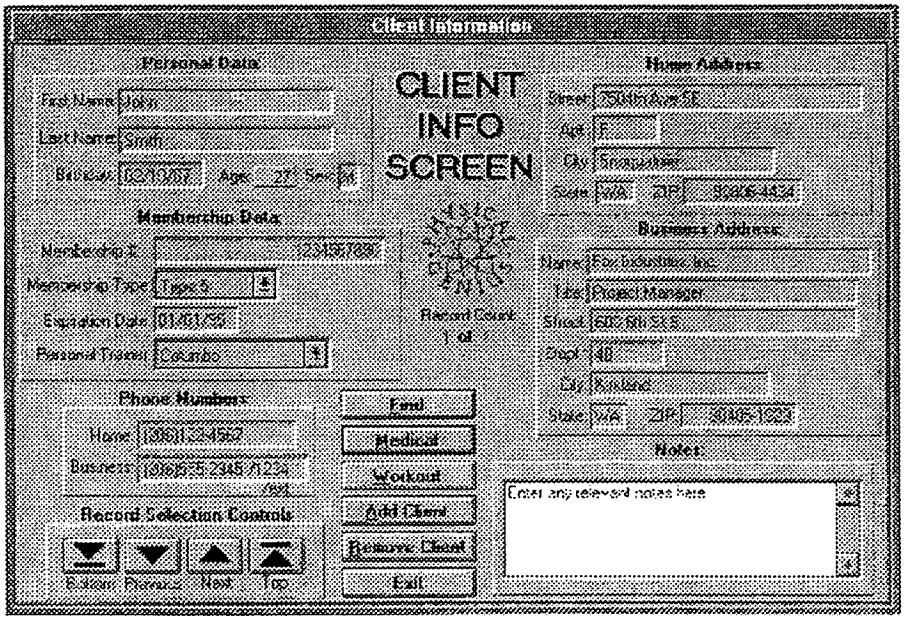

Client Information
CLIENT INFO SCREEN
Personal Data
Membership Data
Phone Numbers
Record Selection Controls
Home Address
Business Address
Notes
Enter any relevant notes here
Find
Medical
Workout
Add Client
Remove Client
Exit

FIG. 27

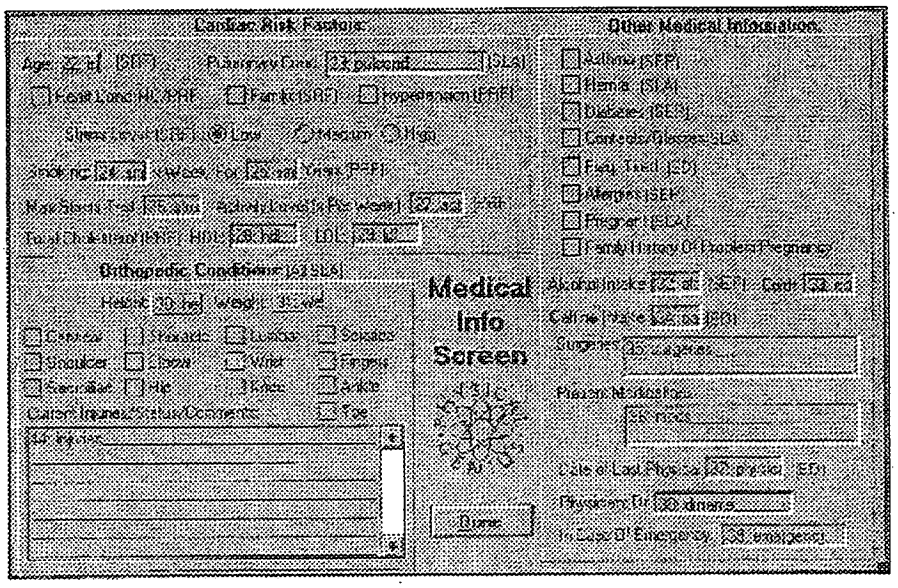

Cardiac Risk Factors
Other Medical Information
Medical Info Screen
Done

FIG. 28

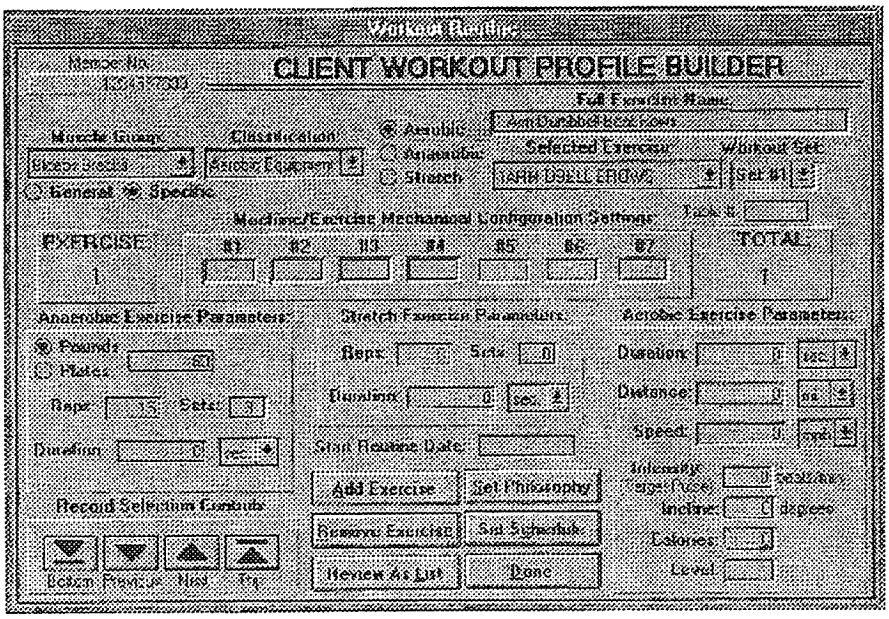

Workout Routines
CLIENT WORKOUT PROFILE BUILDER
Member No.
Full Exercise Name:
Muscle Group
Classification
Aerobic
Anaerobic
Stretch
Selected Exercise:
Workout Set:
General
Specific
Machine/Exercise Mechanical Configuration Settings:
EXERCISE
#1
#2
#3
#4
#5
#6
#7
TOTAL:
Anaerobic Exercise Parameters:
Stretch Exercise Parameters:
Aerobic Exercise Parameters:
Pounds
Meters
Reps:
Sets:
Duration:
Distance:
Speed:
Start Routine Date:
Add Exercise
Set Philosophy
Remove Exercise
Set Schedule
Review As List
Done
Record Selection Controls
Bottom
Previous
Next
Top
Intensity:
Incline:
degrees
Calories:
Level:

FIG. 29

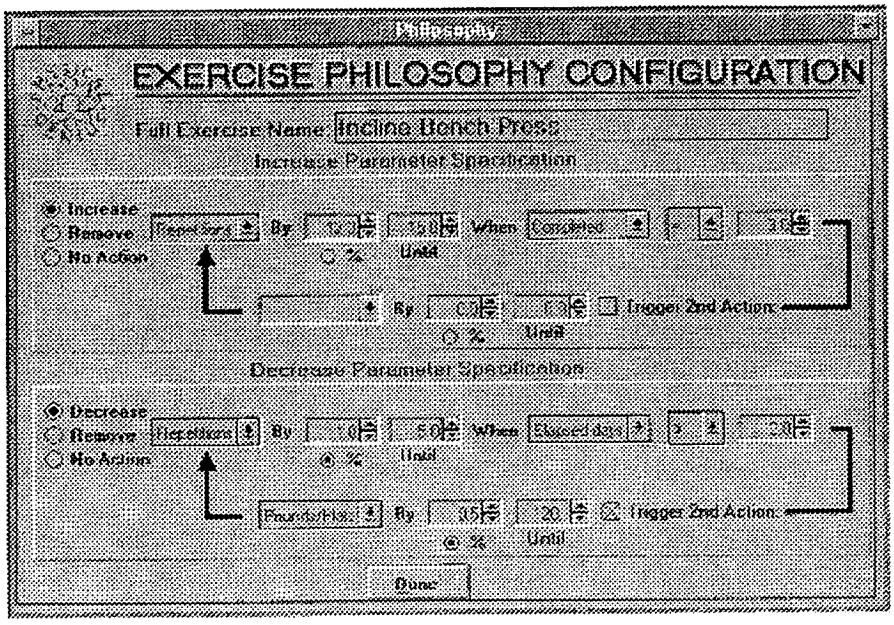

Philosophy
EXERCISE PHILOSOPHY CONFIGURATION
Full Exercise Name: Incline Bench Press
Increase Parameter Specification
Increase
Remove
No Action
By
When
Completed
Until
Trigger 2nd Action
Decrease Parameter Specification
Decrease
Remove
No Action
Repetitions
Exceed data
Done

FIG. 30

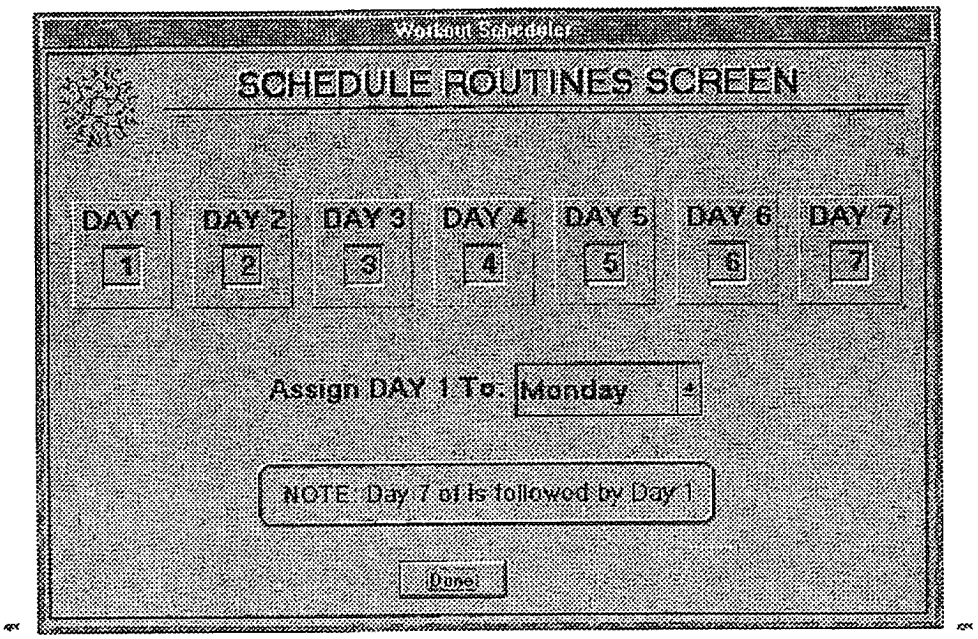

Workout Scheduler
SCHEDULE ROUTINES SCREEN
DAY 1
1
DAY 2
2
DAY 3
3
DAY 4
4
DAY 5
5
DAY 6
6
DAY 7
7
Assign DAY 1 To:
Monday
NOTE: Day 7 of is followed by Day 1
Done!

FIG. 31

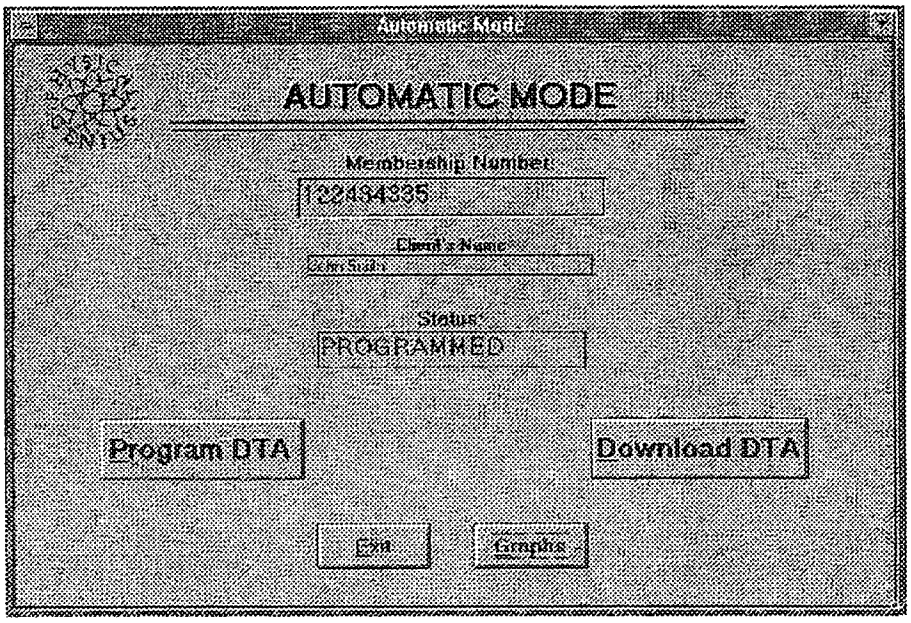

Automatic Mode
AUTOMATIC MODE
Membership Number
12434335
Status
PROGRAMMED
Program DTA
Download DTA
Exit
Graphics

FIG. 32

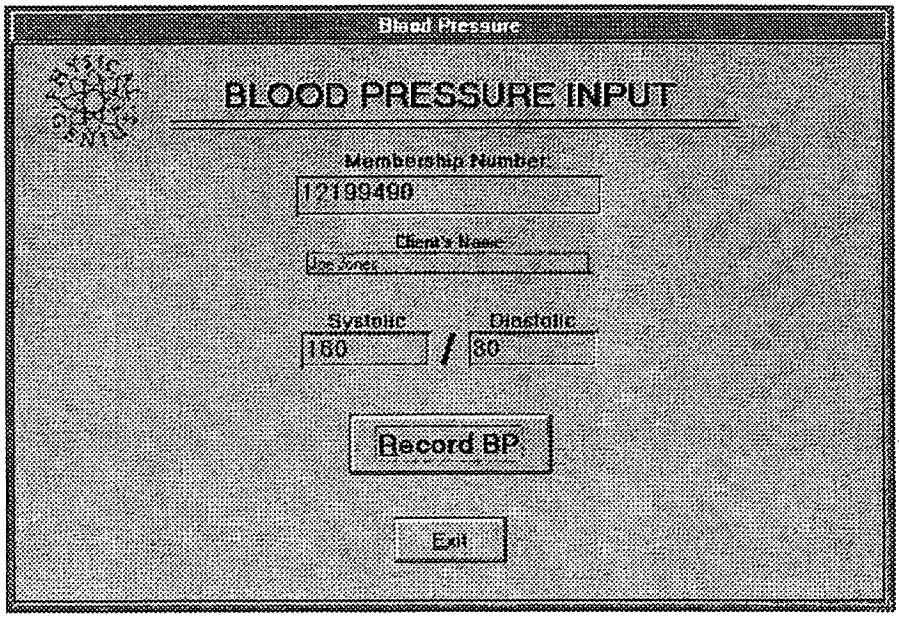

Blood Pressure
BLOOD PRESSURE INPUT
Membership Number
12199490
Client's Name
Systolic
160
/
Diastolic
80
Record BP
Exit

FIG. 33

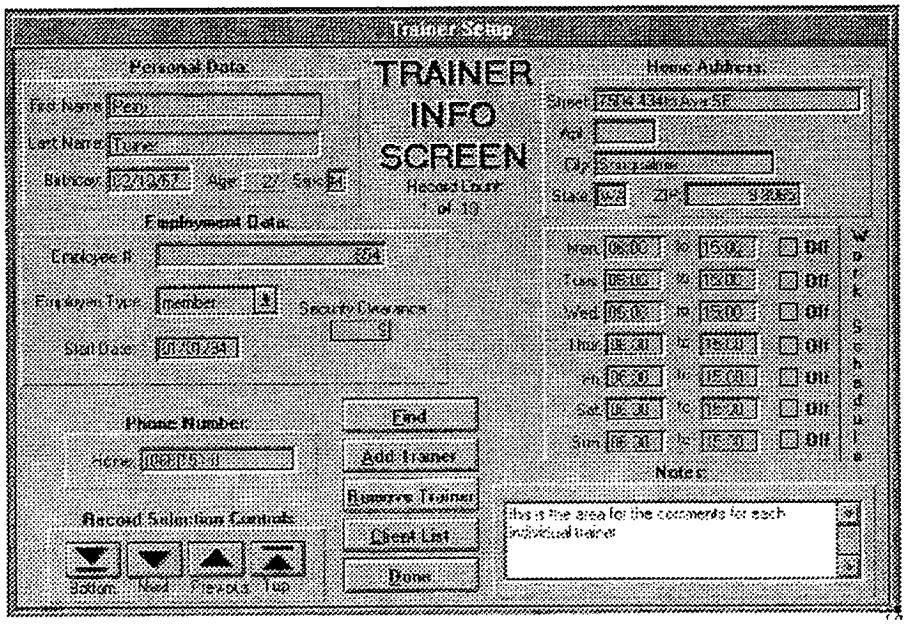

Trainer Setup
Personal Data
TRAINER
INFO
SCREEN
Home Address
Employment Data
Phone Number
Record Selection Controls
Bottom
Next
Previous
Top
Find
Add Trainer
Remove Trainer
Client List
Done
Notes
Work Schedule

FIG. 34

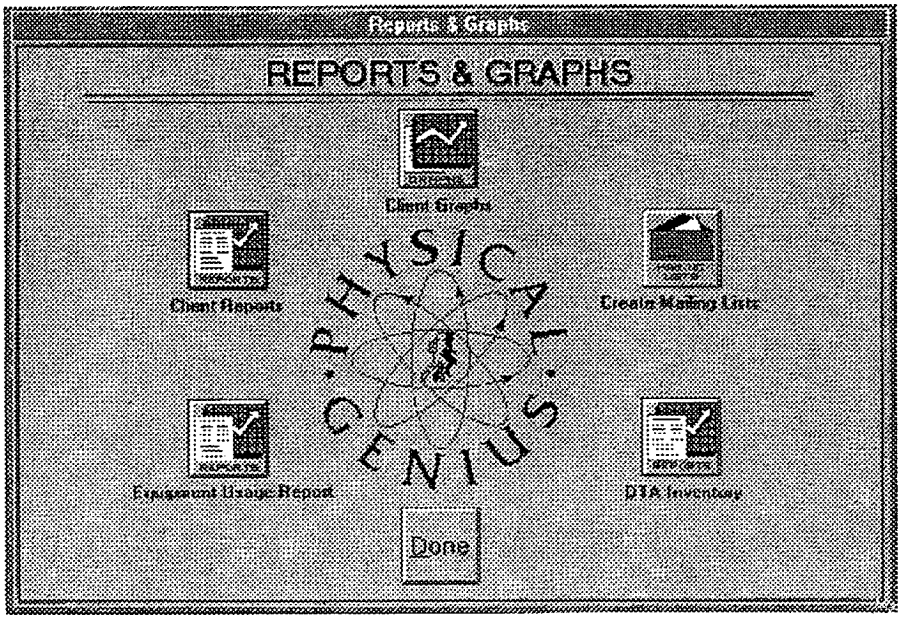

Reports & Graphs
REPORTS & GRAPHS
Client Graphs
Client Reports
Create Mailing Lists
Equipment Usage Report
DTA Inventory
Done

FIG. 35

COMPUTERIZED SYSTEM FOR THE DESIGN, EXECUTION, AND TRACKING OF EXERCISE PROGRAMS

This is a continuation of application Ser. No. 08/285,308 filed on Aug. 9, 1994, now abandoned.

NOTICE REGARDING COPYRIGHTED MATERIAL

FIELD OF THE INVENTION

The invention relates to a computerized exercise fitness program designing system and method for assisting in execution and recording of performance information.

BACKGROUND OF THE INVENTION

A consistent fitness regiment is universally accepted by medical authorities as being instrumental to longevity and good health. There are currently several devices available that monitor an individual exercising on an exercise apparatus. Examples of these are U.S. Pat. Nos. 4,493,485, 4,409,992, and 4,408,183. The problem with these devices is that they are limited to recording the performance the user decides to achieve and provide no instruction on the performance the user should achieve. For example, U.S. Pat. No. 4,493,488 provides a predetermined pace for the user regardless of the user's actual fitness level. Additionally, the aforementioned devices provide no indication of past performance from which a trend can be interpolated. U.S. Pat. No. 4,817,940 does provide the capability to record past performance but makes no provision to automatically calculate future performance requirements based upon that stored information. Additionally U.S. Pat. No. 4,817,940 requires that each piece of exercise equipment be instrumented in order to be accessible by the device.

A key element to a fitness program is the ability to continually challenge the body physically. When a person has performed an exercise at a consistent level, the muscles tend to acclimate to the particular exercise and plateau at the level of development required to maintain the exertion level. In order to advance the development of the muscles involved, the exertion level must be increased at specific intervals. Without continual professional instruction, most amateur fitness enthusiasts lack the basic knowledge to determine when and by how much to advance their workout parameters to achieve maximum effectiveness.

The System described herein allows the Trainer to impart a training philosophy to each exercise. The Trainer can specify up to four actions to be performed, automatically, on specified exercise dependent parameters (pounds of weight, repetitions, sets, etc.) when specified events (completed routines, elapsed days, etc.) occur. The Trainer can have a parameter increase, decrease, or the exercise completely removed from the workout when the specified event occurs. The events are user selectable using dynamically compiled lists and numeric entry fields. Each time client's daily workout routine is requested, the System will compute the new parameters for each exercise using the exercise's philosophy settings, the performance of the last workout session, and the current date. The end result is a workout program that adapts to the client's actual workout performance achieving the desired result without the client possessing a prerequisite knowledge of exercise training or cost of a private trainer each workout session.

The current most popular method of displaying and recording an exercise fitness program involves a cardboard sheet depicting a list of exercises, initial setup parameters, and columns representing workout sessions. The user must manually mark the repetitions, sets, or other work performed for each corresponding exercise. The end result is a very crowded matrix that reveals little trend information and requires manual computation to determine future performance goals. Storage of these sheets may become cumbersome, especially for a large client population.

The System will eliminate the paper workout sheet and replace it with compact portable battery powered computerized Digital Training Assistants (DTAs). Each DTA is dynamically programmed with the user's own fitness routine calculated for the specific workout session. The DTA unit will interactively instruct the user on both the sequence of exercises to be performed as well as the exercises setup and desired performance parameters. A keypad interface to modify parameters is provided to permit recording of the actual work performed. The resultant exercise information is downloaded back to the System where it becomes a permanent part of the user's individual workout performance history database file. The file's data will be used in the computation of the next workout session's exercise performance parameters.

Determining past performance trends using the traditional paper method of parameter recording requires manually graphing data points. This becomes increasingly impractical with increasing number of workouts. The user will often not increase the workout parameters over time or even worse, increment parameters regardless of the actual fitness level attained.

By utilizing the detailed exercise data in the client's individual workout performance history database file, several reports and graphs can be produced with the speed and accuracy of a computerized system. This gives a new level of instant feedback not previously available in the Fitness Club environment. The client is now able to determine the apparent effectiveness of his workout routine and become more connected to his workout by virtue of the near real-time reporting capability of the System. Enthusiasm and involvement are both enhanced.

A common problem with personal trainers at a fitness club is that each trainer will impart their own individual exercise fitness philosophy onto a client. The result being that any two clients trained by different trainers will have differing fitness programs and varied results. There is a loss of standardization within the club and each new trainer adds to the problem. A method of standardizing the training philosophies using a collective expertise is needed within the industry.

The proposed system will allow the Head Trainer to impart his/her training expertise to each Assistant Trainer by mandating that the trainers implement one of the stored exercise routine templates when training a client. The System will allow different exercise templates to be constructed using the equipment and exercises found at the facility. Each template's exercises may have corresponding parameter increasing/decreasing philosophies designed by the Head Trainer as well.

Accuracy in reporting a client's workout performance is another problem encountered by trainers. When a client is not being monitored personally by a trainer, the client is then responsible for remembering and recording the actual work performed. The client must also be aware of the parameters to record and the correct terminology. The Trainer must then be apprised of the client's performance so that he/she may make adjustments as required.

The DTA units allow the client to easily record the actual work performed. Even if a client is unable to complete the exercise as prescribed and displayed, the actual work performed is easily input to the DTA using a numeric keypad and associated function keys. The DTA will prompt the client to make the entry and alert him of any input errors with the required corrective action. The edited workout routine is then downloaded, with computer accuracy, to the System. The changes are instantly noted by the System and adjustments are automatically made.

Currently, at a moderate sized fitness club, the client to trainer ratio is worse than 40 to 1. Most client's are reluctant to incur the additional cost of a Personal Trainer at each session as well as the inconvenience of having to schedule workouts around a trainer's availability. The result is a vast majority of a club's clientele are not taking advantage of the trainers. A potential loss of revenue for the club.

When utilizing the System, the trainers effectively multiply their presence and utility. This is accomplished by having many clients operating the System and DTAs, following the trainer's designed workout program. The automatic parameter philosophy incrementing/decrementing function will ensure that the client is receiving the maximum benefit of the exercise program as if the trainer were always present to make those adjustments in person. An initial trainer/client session for setup and instruction on usage will increase facility's trainer exposure and periodic fitness program checkups ensure further trainer usage.

Currently, a trainer would have to interview and review the workout sheets for several clients to determine if a particular training regiment is performing as prescribed. This is a labor intensive endeavor demanding a large portion of the trainer's work hours. It is to both the trainer's and client's benefit to maximize the effectiveness of a workout routine.

The system will allow the trainer to asses the effectiveness of a workout program by virtue of the system's recording and reporting capabilities combined with the ability to simultaneously look at several different client's and their corresponding demographics. Performance graphs give easily understandable trend analysis useful in determining the effectiveness of the regiment. The trainer may now fine tune the routine to achieve maximum effectiveness.

The comprehensive database capabilities of the system will allow the creation of equipment usage reports charting the number of clients using a particular piece of exercise equipment during a particular week throughout a calendar year this will provide more accurate maintenance scheduling and equipment utilization data.

Demographic and medical information for a client is instantly available to the Trainer as well as the ability to create reports for membership analysis. Such information is very useful in planning promotions and configuration of the facility to better match the client's requirements.

SUMMARY OF THE INVENTION

The System is composed of both a computer software application program and custom computer hardware. The application software, as it currently exists, operates under the Microsoft® Windows™ 3.1 (or above) personal computer operating system. The software constructs and utilizes a variety of Xbase™ type database files to store and retrieve information about clients, Trainers, facility configuration, system preferences, and client workout routines. Computational capabilities include processing exercise workout parameters to produce a dynamic daily workout regiment. Graphical objects are used to implicitly guide the user in operation of the software's embedded functions.

The application software will produce, as one of its prime functions, a workout routine consisting of exercise names and their associated setup and parametirc data. The routine is interactively constructed for a specified client. The routine can then be formatted and electrically transferred to a Digital Training Assistant (DTA). The DTAs utilize a 16-alphanumeric character display and keypad to instruct the user on execution of the workout stored within. The DTAs also allow the various exercise parameters to be modified as needed. Ancillary functions include a stopwatch timer, interactive pulse recording function, and body weight input facility.

The executed exercise routine is eventually transferred from the DTA back to the System where it is reincorporated into the client's database files. The exercise routine is uploaded and downloaded to a DTA by insertion into a special DTA Programming Stand. The DTA Programming Stand is connected to the host computer via an RS-232C interface and converts the electrical signal levels to those compatible with the DTA's own serial interface. Commands from the host computer to the DTA will initiate either the upload or download action. Indicators on the Programming Stand will display power and programming status. Several DTA units may be used with a single DTA Programming Stand all under control of a single software system

BRIEF DESCRIPTION OF THE DRAWINGS

Those and other objects and advantages of the present invention will become more apparent by referring to the following detailed description and accompanying drawings in which:

FIG. 1 Shows a basic diagram of the principal components of the computerized exercise fitness program design, execution, and recording system.

FIG. 2 Shows block diagrams of all versions of the components that constitute the invention.

FIG. 3 Shows a depiction of the Version-A Digital Training Assistant unit.

FIG. 4 Shows a depiction of the Version-A DTA Programming Stand.

FIG. 5 Shows a depiction of the Version-B Digital Training Assistant unit.

FIG. 6 Shows a depiction of the Version-B DTA Programming Stand.

FIG. 7 Shows a detailed circuit diagram of the Version-A Digital Training Assistant unit.

FIG. 8 Shows a detailed circuit diagram of the Version-A DTA Programming Stand.

FIG. 9 Shows a detailed circuit diagram of the Version-B Digital Training Assistant unit.

FIG. 10 Shows a detailed circuit diagram of the Version-B DTA Programming Stand.

FIG. 11 Shows a flow diagram of both Version A & B DTA firmware power-on sequence.

FIG. 12 Shows a flow diagram of both Version A & B DTA firmware NEXT, PREV, ENTER, NAME, PULSE, and WEIGHT keypad key activation sequences

FIG. 13 Shows a flow diagram of the Version-A DTA firmware GROUP keypad key activation sequence.

FIG. 14 Shows a flow diagram of the Version-A DTA firmware DURATION, SETTINGS, PLATES/LBS (DISTANCE), REPS. (SPEED), and SETS (LEVEL) keypad key activation sequences

FIG. 15 Shows a flow diagram of the Version-A DTA firmware TIMER, START/STOP (DURATION), and RESET (CALORIES) keypad key activation sequences.

FIG. 16 Shows a flow diagram of both Version A & B DTA firmware Serial Port Interrupt Routine sequences.

FIG. 17 Shows a flow diagram of the Version-B DTA firmware AEROBIC, ANAEROBIC, and STRETCH keypad key activation sequences

FIG. 18 Shows a flow diagram of the Version-B DTA firmware START/STOP and LAP/RESET keypad key activation sequences

FIG. 19 Shows a flow diagram of the Version-B DTA firmware PLATES/LBS, REPS., SETS, DISTANCE, DURATION, SPEED, INCLINE, LEVEL, and CALORIES keypad key activation sequences

FIG. 20 Shows a connectivity diagram for the application software program depicting screen and file relationships

FIG. 21 Shows a detail of the Flush Mounted Self Adjusting Clip on the Version-B DTA units.

FIG. 22 Shows a detail of the Version-B DTA unit LCD display glass.

FIG. 23 Shows a detailed circuit diagram of the Version-A DTA Unit Keypad.

FIG. 24 Shows a detailed circuit diagram of the Version-B DTA Unit Keypad.

FIG. 25 Administrative Functions Screen representation.

FIG. 26 Facility Exercise Configuration Screen representation.

FIG. 27 Client Information Screen representation.

FIG. 28 Medical Information Screen representation.

FIG. 29 Client Workout Profile Builder Screen representation.

FIG. 30 Exercise Philosophy Configuration Screen representation.

FIG. 31 Schedule Routines Screen representation.

FIG. 32 Automatic Mode Screen representation.

FIG. 33 Blood Pressure Screen representation.

FIG. 34 Trainer Information Screen representation.

FIG. 35 Reports & Graphs Screen representation.

DESCRIPTION OF AN ILLUSTRATIVE EMBODIMENT

A. General Description of the exercise fitness program design, execution, and recording system

Referring to the drawings;

FIG. 1 Shows a basic diagram of the principal components of the computerized exercise fitness program design, execution, and recording system.

Central to the system is a the Personal Computer (PC) **6** that is capable of running Microsoft® Windows™ 3.1 (or above) operating system. The minimum system requirements to run Windows™ also applies to the System's application software. The Keyboard **8** is connected to a 'Y' adapter **5** that also connects the Bar Code or Magnetic Strip reader **9** to the Personal Computer **6**.

The DTA Programming Stand **2** is connected to the PC **6** using an RS-232C serial link cable **3**. The cable **3** can connect to any COM port on the PC **6**. The DTA units **1** are connected to the system by placement into the DTA Programming Stand **2** whereby complimentary electrical contacts from both are engaged.

A mouse pointing device **7** is highly recommended but not essential to the operation of the system application software.

B. Detail Description of the exercise fitness program design, execution, and recording system components

NOTE: It should be noted that there exists two versions of both the Digital Training Assistant units and DTA Programming Stands. Differences will be pointed out where indicated and/or necessary. The System's application software is compatible with both units.

Referring to the drawings;

FIG. 2 Shows block diagrams (both versions) of the components that constitute the invention.

The Version-A DTA Programming Stand interfaces to the host PC via connection to the RS-232C serial link cable through the DB-25 female connector **1**. The signals are then translated to the 3.3 V logic levels required by the DTA units using the EIA/TIA-562 transceiver **2**. Power is supplied by any external regulated power supply capable of delivering 6 VDC@ 400 mA minimum. The power supply must be able to mate with the DC power jack **4**. A SPST switch **3** is used to apply power to Programming Stand. The +6 VDC is routed to a 3.3 V regulator **5** that outputs a steady +3.3 VDC to the remaining circuitry. A green LED **7** will indicate the presence of power to the DTA Programming Stand's internal circuitry. The Version-A DTA interfaces to the Version-A DTA Programming Stand by connecting with DTA Interface Connector **8**. Serial data transmission lines are connected directly to the EIA/TIA-562 transceiver **2**. A red LED **6** connects to a dedicated signal line from the DTA to indicate programming status. The DTA Power Switch Actuator **9** inserts into a hole on the side of the Version-A DTA to activate the DTA's power switch.

The Version-A Digital Training Assistant interfaces to the Version-A DTA Programming Stand via connection with DTA Programmer Interface Connector **10**. The serial and control signals are routed directly to the 8-Bit microcontroller **17**. Power to the DTA unit is supplied by a pair of CR2025 3 V lithium coin cells **14**. A SPST push-on, push-off switch **15** routes power to the dual power regulators **16**. The +3.3 & +5 VDC regulators **16** supply power to the remaining DTA circuitry. The 8-Bit microcontroller **17** interfaces through connector **13** to an OEM LCD Character module **11** & **12**. The OEM LCD Module consists of a +5 VDC LCD Controller **12** and the actual LCD glass **11**. Display contrast is controlled by potentiometer **19**. A 27-Key Keypad **20** connects to the 8-Bit Microcontroller **17** via a pair of connectors **18**. A Piezo Audio Indicator **21** is connected directly to the 8-Bit Microcontroller **17**.

The Version-B DTA Programming Stand interfaces to the host PC via connection to the RS-232C serial link cable through the DB-25 female connector **22**. The signals are then translated to the 3.3 V levels required by the DTA units using the EIA/TIA-562 transceiver **23**. Power is supplied by any external regulated power supply capable of delivering 6 VDC@ 400 mA minimum. The power supply must be able to mate with the DC power jack **30**. A SPST switch **29** is used to apply power to Programming Stand. The +6 VDC is routed to a 3.3 V regulator **28** that outputs a steady +3.3

VDC to the remaining circuitry. A green LED **27** will indicate the presence of power to the DTA Programming Stand's internal circuitry. The Version-B DTA interfaces to the Version-B DTA Programming Stand by connecting with DTA Interface Contacts **25**. Serial data transmission lines are connected directly to the EIA/TIA-562 transceiver **23**. A red LED **24** connects to a dedicated signal line from the DTA to indicate programming status.

The Version-B Digital Training Assistant interfaces to the Version-B DTA Programming Stand via direct contact with DTA Programmer Interface Contacts **35**. The serial and control signals are routed directly to the 8-Bit microcontroller **36**. Power to the DTA unit is supplied by a pair of CR2025 3 V lithium coin cells **39**. A dedicated control signal from the DTA Programming Stand via the DTA Programmer Interface Contacts **35** is used to activate the +3.3 VDC regulator **40**. The +3.3 VDC regulator **40** supplies power to the remaining DTA circuitry. The 8-Bit microcontroller **36** interfaces directly to the T7934 LCD Controller **33**. A ribbon cable **32** connects the row and column drivers to the custom designed +3 VDC 16–5x8 dot matrix character LCD glass **31**. Display contrast is controlled by potentiometer **34**. A 33-Key Keypad **38** connects to the 8-Bit Microcontroller **36** via a ribbon cable **37**. A Piezo Audio Indicator **41** is connected directly to the 8-Bit Microcontroller **36**.

An OEM serial interface cable is used to connect the DTA Programming Stand with the host PC. The cable is composed of a DB-9 female connector **42**, 8 feet of 9 conductor 24-AWG 7x32 STRAND PVC 300 V 80° C. Aluminum Poly Shield cable **43**, and a DB-25 female connector **44**.

Although expressly outside the scope of this document, the host Personal Computer **45** and keyboard emulation type Magnetic Strip or Bar Code reader **46** are included in FIG. **2** for clarity.

C. System Component Mechanical Form

Referring to the drawings:

FIG. **3** Shows a depiction of the Version-A Digital Training Assistant unit.

The ABS plastic case **47** houses the electronic components within. An elastomer keypad **48** is accessed through cutouts in the upper case component. A clear plastic window **49** is used to view the 16-Character LCD dot matrix display **52**. The DTA Programming Stand Interface Connector **50** is visible on the right side view. The power switch actuation hole **51** is also visible next to the connector **50**.

FIG. **4** Shows a depiction of the Version-A DTA Programming Stand.

The ABS plastic case **53** houses the electronic components within. A red LED **54** indicates the status of DTA programming and the green LED **55** indicates the presence of power to the DTA Programming Stand. The DTA is inserted into slot **58** where is connects to the DTA Interface Connector **58A**. The DTA Power Switch Actuator **58B** inserts through the corresponding hole in the side of the DTA unit to activate the internal power switch. The Power Switch **56** is located on the rear panel. The host PC interface DB-25 male connector **57** and DC Power Jack **59** are also located on the rear panel.

FIG. **5** Shows a depiction of the Version-B Digital Training Assistant unit.

The ABS plastic case **60** houses the electronic components within. Elastomer keypads **63**, **71**, **69** & **70** are accessed through cutouts in the case components. The stopwatch control buttons **67** & **68** are separate mechanical switches located on opposing sides of the unit. A clear plastic window **61** is used to view the 16-Character LCD dot matrix display **62**. The flip-open cover **64** reveals the numeric keypad **69** and Function Keypad **70** An integrated Flush Mounted Self-Adjusting Clip **65** is located on the rear of the DTA unit. The DTA Interface Contacts **66** are clearly visible above the cover hinge **60A**.

FIG. **6** Shows a depiction of the Version-B DTA Programming Stand.

The ABS plastic case **72** houses the electronic components within. A red LED **74** indicates the status of DTA programming and the green LED **73** indicates the presence of power to the DTA Programming Stand. The DTA is placed into form fitting depression **75** where it contacts the DTA Interface Contacts **76**. The Power Switch **77** is located on the top of the Programming Stand and is of the push-on, push off type. The host PC interface DB-25 female connector **79** and DC Power Jack **80** are also located on the front panel. Suction cups **78** are provided for secure installation on a smooth flat surface.

FIG. **21** Shows a detail of the Flush Mounted Self Adjusting Clip on the Version-B DTA units.

In Detail 1 of the figure, the force of the compression spring **538** is exceeded by the force of the expansion spring **539** holding the clip in the stowed position. When pressure is applied to the activation depression **543**, as shown in Detail 2, the compression spring **538** is compressed further and the clip pivots on the pivot pin **544**. A ¼" maximum thick surface can now be inserted between the clip and the DTA back surface. When the pressure to the activation depression **543** is released, as sown in Detail 3, The expansion spring **539** is stretched, the clip moves along the pivot slot **545** and the compression spring **538** is allowed to decompress further bringing the clip parallel with the inserted surface. The grip points **542** are now in even contact with the surface providing positive grip. To release the surface, pressure is again applied to the activation depression **543** and the surface is removed.

An alternative design is to replace the expansion spring **539** with a second compression spring **541** located inside the pivot slot **545**. The effect is the same but the design is more compact.

D. System Component Electrical Description

Referring to the drawings:

FIG. **7** Shows a detailed circuit diagram of the Version-A Digital Training Assistant unit.

Power is activated by closing SPST switch **89**. The two CR2025 Lithium Coin Cell Batteries **90** supply +6 VDC to the +5 VDC regulator (LT1121CZ-5) **85** and +3.3 VDC regulator (LT1121CZ-3.3) **86**. Each regulator has a single 1.0 uF filter capacitor **87** & **88**.

The Resistor-Capacitor power-on reset circuit constructed with a diode **82**, 100K resistor **83**, and 1 uF capacitor **84** combine to give an RC constant of 100 mS. The reset signal is sent to the reset input of the 8-Bit Microcontroller **91**.

The 8-bit microcontroller **91** is a Hitachi H8/329 Series single-chip microcomputer (microcontroller). The microcontroller contains 32 Kbytes of Read Only Memory (ROM), 1 Kbyte of Random Access Memory (RAM), one 16-bit free running timer, dual 8-bit timers, a serial communications interface, an A/D converter, and extensive bit controllable I/O ports. The microcontroller **91** operates at +3.3 VDC and utilizes a 4.9152 MHz AT-cut parallel resonating crystal **81** with associated 22 pF capacitors **100** & **101**. A single 0.01 uF decoupling capacitor **92** is provided for the microcontroller **91**.

The microcontroller's **91** P1, P2, and P7 I/O ports are used to receive keypad key press signals via the 15-pin keypad header connectors **94** & **95**. An interrupt signal from the keypad is generated for each key pressed and received by the

microcontroller **91** through the INT0 input (P4-2). Pull-up resistors for those ports not equipped with built-in pull-ups is provided by 27K resistor network **93**.

The microcontroller **91** has a built-in asynchronous serial communications port. The receive and transmit data signals are connected to the DTA Programming Stand Interface Connector **96**. A signal ground is provided to match grounds between DTA and Programming Stand. A program status signal originates from a microcontroller **91** P4-7 port pin to the DTA Programming Stand Interface Connector **96**.

The prerequisite 8-bit data interface to the OEM LCD Display Module is created by using the microcontroller's **91** P3 and P4 I/O ports. The signals are routed directly to the 14-pin LCD header connector **97**. The contrast control signal from the OEM LCD Display Module is available through the header connector **97** and is connected to a **2K** potentiometer **99**. The +5 V supply to the OEM LCD Display module is provided through the header connector **97** as well.

A piezo type audio indicator **98** capable of being driven by a +3.3 V square wave is connected directly to the microcontroller's **91** P4-0 and P4-1 I/O pins (tied together for more drive) to provide audio alarms queues.

FIG. **8** Shows a detailed circuit diagram of the Version-A DTA Programming Stand.

Power to the DTA Programming Stand is supplied by an external AC to DC regulated power supply producing +6 VDC at 400 mA minimum to the DC Power Jack **105**. The power switch **106** routes power to the LT1121CN8-3.3 voltage regulator **107** which supplies +3.3 VDC to the remaining circuitry, A single output 1uF filter capacitor **108** is used. A green LED **115** and associated current limiting resistor **114** indicate the presence of power to the DTA Programming Stand.

The serial data interface to the host PC is provided using the DB-25 female connector **116**. Appropriate control signals are tied together or ground to facilitate asynchronous communication. Transmit and receive data lines are routed to the MAX561CWI +3.3 V Transceiver w/Two EIA/TIA-562 Receivers **111**. The RS-232C signal levels from the host PC are converted to the +3.3 V logical levels for the DTA units and vice versa. Charge-pump 1 uF capacitors **109, 110, 112, & 113** completed the transceiver/converter design.

The DTA interface connector **104** routes serial data transmit and receive signals to the transceiver/converter **111** directly. A common signal ground is provided to equalize voltage levels between the DTA and Programming Stand. A red LED **103** and associated current limiting resistor **102** are used to indicate programming status using a signal from the DTA via the DTA Interface Connector **104**.

FIG. **9** Shows a detailed circuit diagram of the Version-B Digital Training Assistant unit.

Power is activated by a control signal from the DTA Programming Stand received at contact pad **122**. The control signal is connected to the LT1121CS8-3.3 +3.3 VDC regulator **123** shutdown input. A logic high will enable the output of the regulator **123**. The two CR2025 Lithium Coin Cell Batteries **125** supply +6 VDC to the regulator **123**. The regulator has a single 1.0 uF filter capacitor **124**.

The Resistor-Capacitor power-on reset circuit constructed with a diode **119**, 100K resistor **120**, and 1 uF capacitor **121** combine to give an RC constant of 100 mS. The reset signal is sent to the reset input of the 8-bit microcontroller **130**.

The 8-Bit microcontroller **130** is a Hitachi H8/329 Series single-chip microcomputer (microcontroller). The microcontroller contains 32 Kbytes of Read Only Memory (ROM), 1 kbyte of Random Access Memory (RAM), one 16-bit free running timer, dual 8-bit timers, a serial communications interface, an A/D converter, and extensive bit controllable I/O ports. The microcontroller **130** operates at +3.3 VDC and utilizes a 4.9152 MHz AT-cut parallel resonating crystal **127** with associated 22 pF capacitors **128 & 129**. A single 0.01 uF decoupling capacitor **126** is provided for the microcontroller **130**.

The microcontroller's **130** P1, P2, and P7 I/O ports are used to receive keypad key press signals via the keypad ribbon cable connector **137**. An interrupt signal from the keypad is generated for each key pressed and received by the microcontroller **130** through the INT0 input (P4-2). Pull-up resistors for those ports not equipped with built-in pull-ups is provided by 27K resistor network **138**. A diode **139** is used to prevent current from the serial data lines sourced by the DTA Programming Stand to enter the VCC plane of the DTA circuitry.

The microcontroller **130** has a built-in asynchronous serial communications port. The receive and transmit data signals are connected to the DTA Programming Stand Interface Contacts **134 & 135**. A signal ground **136** is provided to match grounds between DTA and Programming Stand. A program status signal originates from the microcontroller **130** port pin to the DTA Programming Stand Interface Contact **133**.

The prerequisite 8-bit data interface to the T7934-0000 LCD Controller **131** is created by using the microcontroller's **130** P3 and P4 I/O ports. The LCD Controller **131** is connected to the 16-Character 5x8 dot matrix +3 V LCD display using a thermal ribbon connector pad(s) **140**. Contrast control is achieved using attached 2 KOhm potentiometer **117**. The T7934-0000 LCD Controller **131** operates using an 120 KOhm oscillation resistor **118**.

A piezo type au indicator **132** capable of being driven by a +3.3 V square wave is connected directly to the microcontroller's **130** P4-0 and P4-1 I/O pins (tied together for added drive) to provide audio alarms queues.

FIG. **10** Shows a detailed circuit diagram of the Version-B DTA Programming Stand.

Power to the DTA Programming Stand is supplied by an external AC to DC regulated power supply producing +6 VDC at 400 mA minimum to the DC Power Jack **141**. The power switch **142** routes power to the LT1121CN8-3.3 regulator **143** which supplies +3.3 VDC to the remaining circuitry, A single output filter 1 uF capacitor **155** is used. A green LED **153** and associated current limiting resistor **152** indicate the presence of power to the DTA Programming Stand.

The DTA Power Control Logic programmable array logic (PAL) device **144** will accept the DTA program status signal from DTA Interface Contact **159** and issues the DTA power control signal to DTA Interface Contact **160**. The 1 KHz oscillator **161** provides a clock signal to operate the logic state machine programmed into the DTA Power Control Logic PAL **144**.

The serial data interface to the host PC is provided using the DB-25 female connector **154**. Appropriate control signals are tied together or ground to facilitate asynchronous communication. Transmit and receive data lines are routed to the MAX561CWI +3.3 V Transceiver w/Two EIA/TIA-562 Receivers **147**. The RS-232C signal levels from the host PC are converted to the +3.3 V logical levels for the DTA units and vice versa. Charge-pump 1 uF capacitors **148, 149, 150, & 151** completed the transceiver/converter design.

The DTA Interface Contacts **156 & 157** route serial data transmit and receive signals to the EIA/TIA-562 transceiver/converter **147** directly. A common signal ground is provided to equalize voltage levels between the DTA and Program-

ming Stand through DTA Interface Contact **158**. A red LED **146** and associated current limiting resistor **145** are used to indicate programming status using a signal from the DTA via the DTA Interface Contact **159**.

FIG. **23** Shows a detailed circuit diagram of the Version-A DTA Unit Keypad.

The elastomer keypad overlays a printed circuit board (PCB) containing electrical traces **166, 167, & 164**. Each keypad protrusion **162** contains a carbon contact pill. The contact pills will cause a short circuit across the printed circuit board traces located under the pill **163** when pressed down onto the PCB. The keypad has a common trace to all key locations that is connected to signal ground **167**. This will electrically ground the other two traces under the contact pill when the key is pressed onto the PCB. A common Key Interrupt signal **164** is routed to all key locations to generate a logic 0 signal with any key press. The individual key actuation signals **166** are routed to a pair of 15 pin header strips **165**.

FIG. **24** Shows a detailed circuit diagram of the Version-B DTA Unit Keypad.

The elastomer keypad overlays a printed circuit board (PCB) containing electrical traces **169, 170, & 171**. Each keypad protrusion **173** contains a carbon contact pill. The contact pills will cause a short circuit across the printed circuit board traces located under the pill **172** when pressed down onto the PCB. The keypad has a common trace to all key locations that is connected to signal ground **171**. This will electrically ground the other two traces under the contact pill when the key is pressed onto the PCB. A common Key Interrupt signal **170** is routed to all key locations to generate a logic 0 signal with any key press. The individual key actuation signals **169** are routed to a ribbon connector **168**. Six of the 33 keys share a common actuation signal path and are discerned by the context in which they are activated.

The START/STOP and LAP/RESET keys are not physically part of the elastomer keypad but their signals are routed the same PCB using separate wires.

FIG. **22** Shows a detail of the Version-B DTA unit LCD display glass.

The Liquid Crystal Display (LCD) glass for the Version-B DTA unit is a custom design utilizing +3 V actuated liquid crystal. The dimensions depicted are also unique to the DTA design.

E. DTA Firmware Description

The microcontroller contains a software program resident in the 32K Read Only Memory (ROM). This program is responsible for the operational interface of the DTA unit encompassing LCD display functions, keypad interface, and serial data transmission/reception. The following paragraphs and figures describe the DTA firmware operation. Where noted, differences between the Version-A and Version-B DTA units are explained

Prior to firmware explanations, a brief discussion of the software interface between the System application software and the DTA is in order.

An exercise routine is formatted into a contiguous digital data stream composed of distinct blocks. The individual bytes of the blocks conform to a strict format as described herein. The routine is compiled by the System application software as the output of a specific function and transmitted to the DTA using the DTA Programming Stand. Once received by the DTA, the DTA's microcontroller and associated firmware program can access, display, and modify the data in the exercise routine as required. The entire modified routine is then eventually transmitted back to the System for reincorporation.

The exercise routine is proceeded by a header block. The block is composed of nineteen bytes describing the client's name and membership number, body weight and pulse measurement. A master Header Command byte determines what mode the DTA is operating under:

00=Normal Mode (normal operation)

02=Learn Mode (allows changes in equipment mechanical/ergonomically settings)

Following the header bytes are the individual Exercise Description Blocks (EDB). The DTA's RAM will support as many exercise blocks as will fit into 960 bytes. The System application software is charged with limiting the formatted exercise routine to this physical limit.

There are six different types of EDBs. These blocks describe the individual exercises that make up the exercise routine. Those blocks specified as having a "Table Lookup" use a code number in place of character bytes that represent the 16-character maximum exercise description displayed by the DTA. A corresponding table of exercise description strings is hardcoded in the DTA's microcontroller's ROM. This method conserves valuable RAM space when an exercise included in the table is used in place of a user defined exercise name.

The following tables describe the formats of the uploaded workout routine data package and associated EDBs:

HEADER BLOCK

BYTE 1	HEADER COMMAND
BYTE 2	MEMBER # LSB
BYTE 3	MEMBER #
BYTE 4	MEMBER #
BYTE 5	MEMBER # MSB
BYTE 6	# OF CHARACTERS IN NAME (0–16)
BYTE 7	222 11111
BYTE 8	4 3333 22
BYTE 9	55555 4444
BYTE 10	77 66666 5
BYTE 11	88888 777
BYTE 12	AAA 99999
BYTE 13	C BBBBB AA
BYTE 14	DDDD CCCC
BYTE 15	FF EEEEE D
BYTE 16	GGGGG FFF
BYTE 17	WEIGHT LSB
BYTE 18	WEIGHT MSB
BYTE 19	PULSE
.	
.	
<Individual Exercise Description Blocks>	
.	
.	
FOOTER BYTE	
CHECKSUM BYTE	

ANAEROBIC EXERCISE DESCRIPTION BLOCK

BYTE 1	COMMAND BYTE	
BYTE 2	BIT 7:	Spare
	BIT 6:	Setting 7 1 = ALPHA 0 = NUMERIC
	BIT 5:	Setting 6 1 = ALPHA 0 = NUMERIC
	BIT 4:	Setting 5 1 = ALPHA 0 = NUMERIC
	BIT 3:	Setting 4 1 = ALPHA 0 = NUMERIC
	BIT 2:	Setting 3 1 = ALPHA 0 = NUMERIC
	BIT 1:	Setting 2 1 = ALPHA 0 = NUMERIC
	BIT 0:	Setting 1 1 = ALPHA 0 = NUMERIC
BYTE 3	BITS 7-5:	# OF SETTINGS
	BITS 4-0:	Setting #1

-continued

ANAEROBIC EXERCISE DESCRIPTION BLOCK		
BYTE 4	333 22222	Setting #2 & #3
BYTE 5	5 44444 33	Setting #3, #4 & #5
BYTE 6	6666 5555	Setting #5 & #6
BYTE 7	XX 77777 6	Setting #6 & #7
	BIT 6:	Spare
	BIT 7:	Completed Status; 0=completed 1=modified
BYTE 8	BIT 7:	0=POUNDS 1=PLATES
	BIT 6:	1 = Decimal in pounds/Plate 0= No decimal
	BITS 5-0:	POUNDS/PLATES MSB
BYTE 9	BITS 7-0	POUNDS/PLATES LSB
BYTE 10	BITS 7-4	Duration type 0 = seconds 1 = minutes 2=hours
	BITS 3-0	Spare
BYTE11	BITS 7 - 0	DURATION
BYTE 12	BITS 7 - 0	REPS
BYTE 13	BITS 7-4:	SETS
	BITS 3-0	#CHARACTERS IN EQUIPMENT NAME(1-16)+1
BYTE 14	222 11111	
BYTE 15	4 33333 22	
BYTE 16	5555 4444	
BYTE 17	77 66666 5	
BYTE 18	88888 777	
BYTE 19	AAA 99999	
BYTE 20	C BBBBB AA	
BYTE 21	DDDD CCCC	
BYTE 22	FF EEEEE D	
BYTE 23	GGGGG FFF	

ANAEROBIC W/TABLE LOOKUP EXERCISE DESCRIPTION BLOCK		
BYTE 1	COMMAND BYTE	
BYTE 2	BIT 7:	Spare
	BIT 6:	Setting 7 1 = ALPHA 0 = NUMERIC
	BIT 5:	Setting 6 1 = ALPHA 0 = NUMERIC
	BIT 4:	Setting 5 1 = ALPHA 0 = NUMERIC
	BIT 3:	Setting 4 1 = ALPHA 0 = NUMERIC
	BIT 2:	Setting 3 1 = ALPHA 0 = NUMERIC
	BIT 1:	Setting 2 1 = ALPHA 0 = NUMERIC
	BIT 0:	Setting 1 1 = ALPHA 0 = NUMERIC
BYTE 3	BITS 7-5:	# OF SETTINGS
	BITS 4-0:	Setting #1
BYTE 4	333 22222	Setting #2 & #3
BYTE 5	5 44444 33	Setting #3, #4 & #5
BYTE 6	6666 5555	Setting #5 & #6
BYTE 7	XX 77777 6	Setting #6 & #7
	BIT 6:	Spare
	BIT 7:	Completed Status; 0=completed 1=modified
BYTE 8	BIT 7:	0=POUNDS 1=PLATES
	BIT 6:	1 = Decimal in pounds/Plate 0= No decimal
	BITS 5-0:	POUNDS/PLATES MSB
BYTE 9	BITS 7-0	POUNDS/PLATES LSB
BYTE 10	BITS 7-0	REPS
BYTE11	BITS 7-4:	SETS
	BITS 3-0	Duration type 0=seconds 1=minutes 2=hours
BYTE 12	BITS 7 - 0	DURATION
BYTE 13	LOOKUP TABLE # LSB	
BYTE 14	LOOKUP TABLE # MSB	

AEROBIC EXERCISE DESCRIPTION BLOCK		
BYTE 1	COMMAND BYTE	
BYTE 2	BIT 7:	Spare
	BIT 6:	Setting 7 1 = ALPHA 0 = NUMERIC
	BIT 5:	Setting 6 1 = ALPHA 0 = NUMERIC
	BIT 4:	Setting 5 1 = ALPHA 0 = NUMERIC
	BIT 3:	Setting 4 1 = ALPHA 0 = NUMERIC
	BIT 2:	Setting 3 1 = ALPHA 0 = NUMERIC
	BIT 1:	Setting 2 1 = ALPHA 0 = NUMERIC
	BIT 0:	Setting 1 1 = ALPHA 0 = NUMERIC
BYTE 3	BITS 7-5:	# OF SETTINGS
	BITS 4-0:	Setting #1
BYTE 4	333 22222	Setting #2 & #3
BYTE 5	5 44444 33	Setting #3, #4 & #5
BYTE 6	6666 5555	Setting #5 & #6
BYTE 7	XX 77777 6	Setting #6 & #7
	BIT 6:	Spare
	BIT 7:	Completed Status; 0=completed 1=modified
BYTE 8	BIT 7-0:	INTENSITY beats per minute
BYTE 9	BITS 7:	1=LEVEL IS ALPHA 0=LEVEL IS NUMERIC
	BITS 6-0:	LEVEL
BYTE 10	BITS 7-5:	RATE: 0=MPH 1=KPH 2=FPM 3=MPM 4=SPM 5 -7=sp
	BITS 4-0:	INCLINE
BYTE 11	BIT 7	SPEED Decimal Indicator 1 = decimal point 0 = no decimal
	BITS 6-0	SPEED MSB
BYTE 12	BITS 7-0	SPEED LSB
BYTE 13	BITS 7-0:	CALORIES x 10
BYTE 14	BIT 7-6:	Distance Type 0= Miles 1= feet 2= Kilometers 3= Meters
	BIT 5	Distance Decimal Indicator 1 = decimal present
	BITS 4-0	DISTANCE MSB
BYTE 15:	BITS 7-0:	DISTANCE LSB
BYTE 16	BITS 7-0:	DURATION
BYTE 17:	BITS 7-4:	Duration type 0 = seconds 1=minutes 2=hours
	BITS 3-0	# CHARACTERS IN EQUIPMENT NAME(1-16)+1
BYTE 18	222 11111	
BYTE 19	4 33333 22	
BYTE 20	5555 4444	
BYTE 21	77 66666 5	
BYTE 22	88888 777	
BYTE 23	AAA 99999	

-continued

AEROBIC EXERCISE DESCRIPTION BLOCK	
BYTE 24	C BBBBB AA
BYTE 25	DDDD CCCC
BYTE 26	FF EEEEE D
BYTE 27	GGGGG FFF

AEROBIC W/TABLE LOOKUP EXERCISE DESCRIPTION BLOCK		
BYTE 1	COMMAND BYTE	
BYTE 2	BIT 7:	Spare
	BIT 6:	Setting 7 1 = ALPHA 0 = NUMERIC
	BIT 5:	Setting 6 1 = ALPHA 0 = NUMERIC
	BIT 4:	Setting 5 1 = ALPHA 0 = NUMERIC
	BIT 3:	Setting 4 1 = ALPHA 0 = NUMERIC
	BIT 2:	Setting 3 1 = ALPHA 0 = NUMERIC
	BIT 1:	Setting 2 1 = ALPHA 0 = NUMERIC
	BIT 0:	Setting 1 1 = ALPHA 0 = NUMERIC
BYTE 3	BITS 7-5:	# OF SETTINGS
	BITS 4-0:	Setting #1
BYTE 4	333 22222	Setting #2 & #3
BYTE 5	5 44444 33	Setting #3, #4 & #5
BYTE 6	6666 5555	Setting #5 & #6
BYTE 7	XX 77777 6	Setting #6 & #7
	BIT 6:	Spare
	BIT 7:	Completed Status; 0=completed 1=modified
BYTE 8	BIT 7-0:	INTENSITY beats per minute
BYTE 9	BITS 7:	1=LEVEL IS ALPHA 0=LEVEL IS NUMERIC
	BITS 6-0:	LEVEL
BYTE 10	BITS 7-5:	RATE: 0=MPH 1=KPH 2=FPM 3=MPM 4=SPM 5 -7=sp
	BITS 4-0:	INCLINE
BYTE 11	BIT 7	SPEED Decimal Indicator 1= decimal point 0 = no decimal
	BITS 6-0	SPEED MSB
BYTE 12	BITS 7-0	SPEED LSB
BYTE 13	BITS 7-0:	CALORIES x 10
BYTE 14	BIT 7-6:	Distance Type 0= Miles 1= feet 2= Kilometers 3= Meters
	BIT 5	Distance Decimal Indicator 1 = decimal present
	BITS 4-0	DISTANCE MSB
BYTE 15:	BITS 7-0:	DISTANCE LSB
BYTE 16:	BITS 7-0:	DURATION
BYTE 17:	BITS 7-4:	Duration type 0 = seconds 1=minutes 2=hours
	BITS 3-0:	Spare
BYTE 18	LOOKUP TABLE # LSB	
BYTE 19	LOOKUP TABLE # MSB	

STRETCH EXERCISE DESCRIPTION BLOCK		
BYTE 1	COMMAND BYTE	
BYTE 2	BITS 7-0:	DURATION
BYTE 3	BITS 7-4:	Duration type 0 = seconds 1=minutes 2=hours
	BITS 3-1:	Spare
	BIT 0:	Completed Status; 0= completed 1=modified
BYTE 4	BIT 7-0:	REPS
BYTE 5	BITS 7-4:	SETS
	BITS 3-0	# CHARACTERS IN EQUIPMENT NAME(1-16)+1
BYTE 6	222 11111	
BYTE 7	4 33333 22	
BYTE 8	5555 4444	
BYTE 9	77 66666 5	
BYTE 10	88888 777	
BYTE 11	AAA 99999	
BYTE 12	C BBBBB AA	
BYTE 13	DDDD CCCC	
BYTE 14	FF EEEEE D	
BYTE 15	GGGGG FFF	

STRETCH W/TABLE LOOKUP EXERCISE DESCRIPTION BLOCK		
BYTE 1	COMMAND BYTE	
BYTE 2	BITS 7-0:	DURATION
BYTE 3	BITS 7-4:	Duration type 0 = seconds 1=minutes 2=hours
	BITS 3-1:	Spare
	BIT 0:	Completed Status; 0= completed 1=modified
BYTE 4	BIT 7-0:	REPS
BYTE 5	BITS 7-4:	SETS
	BITS 3-0	not used
BYTE 6	LOOKUP TABLE # LSB	
BYTE 7	LOOKUP TABLE # MSB	

FIG. 11 Shows a flow diagram of both Version A & B DTA firmware power-on sequence.

When power is applied to the DTA's microcontroller, the microcontroller's program execution unit will reset and begin execution at Power On 174. The program then initializes the microcontroller 175 setting up various internal registers and the interrupt vector table. Next the program will initialize the OEM LCD Display Module's T7934 LCD Controller 176.

Following initialization, the program will perform some internal self tests. The registers located in the microcontrol-

ler are tested for both read and write capability **177**. If the any part of the test fails **178**, the alarm is sounded and a message is displayed on the LCD with a specific code number indicating the failed register **179**. The program jumps to an infinite loop **180** causing a general program abort. The DTA must be powered off then on to reset.

The 1 Kbyte RAM is tested next **181** using seven different bit patterns written to and then read from all the RAM locations. If an error is detected at any stage of the test **182**, the alarm is sounded and a message is displayed on the LCD with a specific code number indicating the failed location **183**. The program jumps to an infinite loop **184** causing a general program abort. The DTA must be powered off then on to reset.

The keypad connected to the microcontroller via the I/O ports is tested to determine if any keys are stuck active **185**. If a key is detected as being stuck on **186**, the alarm is sounded and a special message is displayed on the LCD with the offending key number **187**. The test repeats until the key is unstuck.

Once the testing has completed successfully, the start message is displayed: "Physical Genius" **189** followed by: "Ready to Program" message. The program enters a loop **190** awaiting the reception of a exercise routine data upload.

While in the waiting loop **190**, the program checks for any receiver errors **191**. If detected, the alarm is sounded and a special message is displayed **192** describing the offending error. The routine will reset awaiting reception of more data.

When a valid upload of a workout routine is received, a new message is displayed **193** instructing the user to advance the program with the NEXT key. The program now enters a waiting loop **195** looking for a NEXT key activation. Upon detection of the key, the program will process the keypad key request **194**.

From this point on, the program is event driven. Keypad requests will determine what function the program executes next and further keypad requests control the sub-functionality therein.

When a key is activated, an interrupt signal is generated to the microcontroller which activates a special interrupt processing routine. The keyboard I/O port pins are scanned and the activated key's signal is detected. The keyboard interrupt routine will return a number representing the key that was detected as being active.

FIG. 12 Shows a flow diagram of both Version A & B DTA firmware NEXT, PREV, ENTER, NAME, PULSE, and WEIGHT keypad key activation sequences.

When the NEXT Key has been detected **196**, the program will locate the next exercise description block **197**. If while searching for the next EDB, the program encounters the Footer byte indicating the end of the upload block **198**, a special message is displayed **199** instructing the user to hit the PREV key. The program waits until the PREV key is hit **201**. When detected the program will execute the display sequence routine **200** for the EDB that was being displayed prior to the NEXT key activation.

The exercise display sequence routine **200** will parse out the elements of the exercise description block (EDB) and display them on the LCD display. Where indicated, if the associated exercise parameter is 0, the parameter's display will be omitted. The following tables depict the display sequence for each main type of exercise:

STRETCH Excerises:	ANAEROBIC Exercises:	AEROBIC Exercises:
Name of Exercise	Name of Exercise	Name of Exercise
delay 3 seconds	delay 3 seconds	delay 3 seconds
Repetitions Setting	Mechanical/	Mechanically/
(if not 0)	Ergonomically	Ergonomically
delay 3 seconds	Settings	Settings
Sets Setting	delay 3 seconds	delay 3 seconds
(if not 0)	Pounds/Plates Setting	Duration Setting
delay 3 seconds	(if not 0)	(if not 0)
	delay 3 seconds	delay 3 seconds
	Repetitions Setting	Distance Setting
	(if not 0)	(if not 0)
	delay 3 seconds	delay 3 seconds
	Sets Setting	Speed Setting
	(if not 0)	(if not 0)
	delay 3 seconds	delay 3 seconds
	Duration Setting	Intensity Setting
	(if not 0)	(if not 0)
	delay 3 seconds	delay 3 seconds
		Calorie Setting
		(if not 0)
		delay 3 seconds
		Incline Setting
		(if not 0)
		delay 3 seconds
		Level Setting
		(if not 0)
		delay 3 seconds

The sequence is repeated three times **202** after which the microprocessor will enter its low power sleep mode **203** until a keypad key is pressed and thus the activation interrupt is detected. The sequence will then restart at the top. Any time during the display sequence, the program will be able to process keypad key activation interrupts and act accordingly.

When the PREV Key has been detected **204**, the program will locate the previous EDB **205**. If while searching for the previous exercise description block, the program advances beyond the beginning of the exercise description space **206**, a special message is displayed **207** instructing the user to hit the NEXT key. The program waits until the NEXT key is hit **209**. When detected the program will continue to the display sequence routine of the EDB being displayed prior to activating the PREV key **208**.

The sequence is repeated three times **210** after which the microprocessor will enter its low power sleep mode **211** until a keypad key is pressed and thus the activation interrupt is detected. The sequence will then restart at the top.

If the ENTER key is detected **212**, the program must first decide the context of the key's activation **213**. If the key is used as data input confirmation, the data is then validated according to the context of the data entry **215**. If there is a detected entry error **216**, a special message is displayed indicating the error and corrective action **217**. The program then repeats the original data entry prompt routine **218** followed by another ENTER key detection **212**. If the data was determined as valid and correct it is formatted and stored in the appropriate location in the current EDB or Header Block location **219**. The program then continues from where it was originally interrupted or made the function call **220**.

If the context of the ENTER key was not as data input confirmation then it is used to mark the current EDB as being completed **214**. The program then continues at Marker A.

If the NAME key is detected **221**, the client's name characters are extracted and reconstructed as ASCII characters **222** from the Header Block bytes 7 through 16. The bytes are then sent to the LCD Display Controller for a ten

second display 223. Following the interval, the program returns to the point of interruption 224.

If the WEIGHT key is detected 225, the weight entry function is activated with the original weight value stored in the uploaded Header Block extracted 226 from bytes 17 and 18. The bytes are converted to ASCII and sent to the LCD Display Controller for display 227. The program then prompts for a new entry 228 allowing a three digit plus one decimal place value. The program continues at Marker B for ENTER key processing 229. When successfully concluded, the program will return to the point of interruption 230.

If the PULSE key is detected 231, the pulse entry function is activated. The function will interactively instruct, via the LCD display, the user on how and when to measure his pulse 232. The program will then start a ten second timer 233. At the conclusion of the time interval, the alarm sounds and the program prompts for the new measured pulse count entry 234. The program continues at the Marker B for ENTER key processing 235. When successfully concluded, the program will return to the point of interruption 236.

FIG. 13 Shows a flow diagram of the Version-A DTA firmware GROUP keypad key activation sequence.

The GROUP key is used to advance the exercise program out of the predetermined sequence to the next available exercise in the specified group (Aerobic Anaerobic, or Stretch). If the GROUP key is detected 237, the word "Aerobic" is displayed 238. A three second timer is started 239 and if the GROUP key is detected again 240 within the interval, the program will locate the first available Aerobic type EDB 241. The display sequence is activated at Marker C, 247. If no uncompleted Aerobic EDB is found 243, a special error display message is used 245 and the program returns to the point of interruption 248.

If the three second timer is allowed to elapse 242, the word "Anaerobic" is displayed 244. A three second timer is started 246 and if the GROUP key is detected again 249 within the interval, the program will locate the first available Anaerobic type EDB 250. The display sequence is activated at Marker C 256. If no uncompleted Anaerobic EDB is found 252, a special error display message is used 254 and the program returns to the point of interruption 257.

If the three second timer is allowed to elapse 251, the word "Stretch" is displayed 253. A three second timer is started 255 and if the GROUP key is detected again 258 within the interval, the program will locate the first available Stretch type EDB 259. The display sequence is activated at Marker C 263. If no uncompleted Stretch EDB is found 261, a special error display message is used 262 and the program returns to the point of interruption 254.

If the three second timer is allowed to elapse 260, the sequence starts over again with displaying "Aerobic" 238.

FIG. 14 Shows a flow diagram of the Version-A DTA firmware DURATION, SETTINGS, PLATES/LBS (DISTANCE), REPS. (SPEED), and SETS (LEVEL) keypad key activation sequences

In the Version-A DTA unit firmware, selected keys perform dual functions depending upon the context in which they are activated

If the DURATION key is detected 265, the program will extract the duration setting from the current EDB block 266 and allow a new input 267. The program continues at B Marker for ENTER key processing 268. When successfully concluded, the program will return to the point of interruption 269

If the SETTINGS key is detected 270, the program will first check if the current mode is Learn and that the EDB is not a Stretch type exercise. If otherwise, the function is not entered. If true, the program will extract the mechanical/ergonomically settings 271 from the EDB and display the first setting 272. The program will now allow input of a new setting 273. The program continues at Marker B for ENTER key processing 274. When successfully concluded, the program will check to see if there is another setting to adjust 275. If so, the process is repeated for the remaining settings 272. If finished, the program will return to the point of interruption 276

If the PLATES/LBS (DISTANCE) key is detected 277, the program will first determine the context by examining the EDB for Aerobic type exercise 278. If Aerobic, the program will extract and display the Distance value 279. The program next allows the input of a new distance setting 281. The program continues at Marker B for ENTER key processing 283. When successfully concluded, the program will return to the point of interruption 285 If the EDB was Stretch 287, a special error message is displayed 288 and the program will return to the point of interruption 289. If the program determines the EDB is Anaerobic, the program will extract and display the Pounds/Plates value 280. The program next allows the input of a new Pounds/Plates setting 282. The program continues at Marker B for ENTER key processing 284. When successfully concluded, the program will return to the point of interruption 286

If the REPS. (SPEED) key is detected 290, the program will first determine the context by examining the EDB for Aerobic type exercise 291. If Aerobic, the program will extract and display the Speed value 292. The program next allows the input of a new speed setting 294. The program continues at Marker B for ENTER key processing 296. When successfully concluded, the program will return to the point of interruption 298

If the program determines the EDB is not Aerobic, the program will extract and display the Repetitions value 293. The program next allows the input of a new Repetitions setting 295. The program continues at Marker B for ENTER key processing 297. When successfully concluded, the program will return to the point of interruption 299

If the SETS. (LEVEL) key is detected 300, the program will first determine the context by examining the EDB for Aerobic type exercise 301. If Aerobic, the program will extract and display the Level value 302. The program next allows the input of a new level setting 304. The program continues at Marker B for ENTER key processing 306. When successfully concluded, the program will return to the point of interruption 308

If the program determines the EDB is not Aerobic, the program will extract and display the Sets value 303. The program next allows the input of a new Sets setting 305. The program continues at Marker B for ENTER key processing 307. When successfully concluded, the program will return to the point of interruption 309

FIG. 15 Shows a flow diagram of the Version-A DTA firmware TIMER, START/STOP (DURATION), and RESET (CALORIES) keypad key activation sequences.

If the TIMER key is detected 310, the program enters the count down timer function. The program first displays the "Set Minutes" prompt 311. Entry of a minutes setting is enabled 312. The program continues at Marker B for ENTER key processing 313. When successfully concluded, the program will now display the "Set Seconds" prompt 314. Entry of a seconds setting is enabled 315. The program continues at Marker B for ENTER key processing 316. When successfully concluded, the program now waits for a keypad key activation 317. If the START/STOP (INCLINE) key is detected the timer and display time function is started 320.

For other detected keys, the program checks if it was a numeric key **318**, if so the program ignores the key activation **317**, If it is a function key, the program will return to the point of interruption **319**.

After the timer and display time function has started **320**, the program will also look for a key activation **321**. Any other key but the START/STOP (INCLINE) key will be ignored. If the START/STOP (INCLINE) key is detected, the timer and time display are stopped **322**. If the TIMER key is detected again **323**, the program will continue at Marker E. If the RESET (CALORIES) key is detected, the timer is reset to the initial values **326** and program execution continues at **317**. If the START/STOP (INCLINE) key is detected, the program execution continues at Marker D.

In the absence of the START/STOP (INCLINE) key detection, the timer will continue to count down to 00:00 while displaying the current remaining time. When the timer has reach 5 remaining seconds **327**, the alarm will beep on each subsequent second **328**. When the timer expires **329**, the alarm will beep several times and the timer stop.

At this point, the program is waiting for another key activation **331**. If the TIMER key is detected, the program continues execution at Marker E. If the RESET (CALORIES) key is detected, the program will reset the timer to the initial values **326** and continue execution at **317**. If a numeric key was detected **332**, the key activation is ignored. If it was a function key, the program will return to the point of interruption **333**.

The START/STOP (INCLINE) and RESET (CALORIES) keys have dual functions. Following are descriptions of their alternative functions

If the START/STOP (INCLINE) key is detected **334**, the program determines what context it is currently in **336**. If the timer is active, the timer functionality is used **335**. If not in timer mode, the program checks to see if the current EDB is Aerobic **337**. If not then a special error message is displayed **339** and the program will return to the point of interruption **341**.

For an Aerobic EDB, the program will extract and display the Incline value **338**. The program next allows the input of a new incline setting **340**. The program continues at Marker B for ENTER key processing **342**. When successfully concluded, the program will return to the point of interruption **343**

If the RESET (CALORIES) key is detected **344**, the program determines what context it is currently in **346**. If the timer is active, the timer functionality is used **345**. If not in timer mode, the program checks to see if the current EDB is Aerobic **347**. If not then a special error message is displayed **349** and the program will return to the point of interruption **351**.

For an Aerobic EDB, the program will extract and display the Calories value **348**. The program next allows the input of a new calories setting **350**. The program continues at Marker B for ENTER key processing **352**. When successfully concluded, the program will return to the point of interruption **353**

FIG. **16** Shows a flow diagram of both Version A & B DTA firmware Serial Port Interrupt Routine sequences.

The microcontroller's serial communications interface will generate an interrupt when a byte has been received **354**. The interrupt routine will check to see if any receive errors were accrued **355**. If so a special error message is displayed **356** and the program returns to the point of interruption **374**.

If there are no errors, the byte is examined to see if it is a download command **357**. If so the processing continues to process a download starting at **375**. If not then the program checks to see if a previous upload has been received **358**. If so, the interrupt routine is exited and the program returns to the point of interruption **374**.

If no previous upload has been received, the program checks to see if the byte was a valid master header byte **359**. If not then display special error message and the program returns to the point of interruption **374**.

If a valid master header byte is detected, the remaining header block bytes are received and put into a buffer in RAM **361**. After the header bytes have been received, the program will examine the next byte received and check if it is an Aerobic EDB header byte **362**, if so the program will determine how many more bytes in the block, receive them, and store in RAM buffer **363**.

If the EDB header byte was not Aerobic, the program checks to see if it was Anaerobic **364**. if so the program will determine how many more bytes in the block, receive them, and store in RAM buffer **365**.

If the EDB header byte was not Anaerobic, the program checks to see if it was Stretch **366**. if so the program will determine how many more bytes in the block, receive them, and store in RAM buffer **367**.

If the EDB header byte was not Stretch, the program checks to see if it was the Footer byte **368**. if not the program will display a special error message **383** and returns to the point of interruption **374**.

If the Footer byte is detected, the entire received buffer is modulo-256 checksummed **369**. The result is compared to the embedded checksum value that followed the Footer byte in the block **370**. If there was a mismatch, the program displays a special error message **371**, sets a failure flag **373** and returns to the point of interruption **374**. If the checksum test passes, the received valid flag is set **372** and the program returns to the point of interruption **374**.

For the download command received case, the entire exercise routine block (from master header byte to Footer byte) is modulo-256 checksummed **375** and the result appended as the last byte in the block **376**. The entire block is then transmitted out the serial interface one byte at a time **377**. When the transmitter is detected as empty **378**, the next byte is transmitted. When the last byte has been detected **379**, the byte is transmitted **381** and the program returns to the point of interruption **382**.

FIG. **17** Shows a flow diagram of the Version-B DTA AEROBIC, ANAEROBIC, and STRETCH keypad key activation sequences

The Version-B DTA unit expands the Version-A DTA's GROUP key into the three separate exercise groups selection buttons; Aerobic, Anaerobic, and Stretch

If the AEROBIC key is detected **384**, the program will start at the first EDB and searches for the first uncompleted Aerobic exercise **385**. If no uncompleted aerobic exercise is found **386**, a special error message is displayed **387** and the program returns to the point of interruption **389**.

If an uncompleted Aerobic exercise is found, the exercise display sequence routine **388** will parse out the elements of the EDB and display them on the LCD. If the exercise parameter is 0, its display may be omitted. The display sequence is as follows:

AEROBIC Excerises:
Name of Exercise
delay 3 seconds
Mechanically/Ergonomically Settings

-continued

AEROBIC Excerises:
delay 3 seconds
Duration Setting (if not 0)
delay 3 seconds
Distance Setting (if not 0)
delay 3 seconds
Speed Setting (if not 0)
delay 3 seconds
Intensity Setting (if not 0)
delay 3 seconds
Calorie Setting (if not 0)
delay 3 seconds
Incline Setting (if not 0)
delay 3 seconds
Level Setting (if not 0)
delay 3 seconds

The sequence is repeated three times **390** after which the microprocessor will enter its low power sleep mode **391** until a keypad key is activated and the resultant activation interrupt is detected. The sequence will then restart at the top.

If the ANAEROBIC key is detected **392**, the program will start at the first EDB and searches for the first uncompleted Anaerobic exercise **393**. If no uncompleted anaerobic exercise is found **394**, a special error message is displayed **395** and the program returns to the point of interruption **396**.

If an uncompleted Anaerobic exercise is found, the exercise display sequence routine **397** will parse out the elements of the exercise description block and display them on the LCD. If the exercise parameter is 0, its display may be omitted. The display sequence is as follows:

ANAEROBIC Exercises:
Name of Exercise
delay 3 seconds
Mechanical/Ergonomically Settings
delay 3 seconds
Pounds/Plates Setting (if not 0)
delay 3 seconds
Repetitions Setting (if not 0)
delay 3 seconds
Sets Setting (if not 0)
delay 3 seconds
Duration Setting (if not 0)
delay 3 seconds

The sequence is repeated three times **398** after which the microprocessor will enter its low power sleep mode **399** until a keypad key is activated and the resultant activation interrupt is detected. The sequence will then restart at the top.

If the STRETCH key is detected **400**, the program will start at the first EDB and searches for the first uncompleted Stretch exercise **401**. If no uncompleted stretch exercise is found **402**, a special error message is displayed **403** and the program returns to the point of interruption **404**.

If an uncompleted Stretch exercise is found, the exercise display sequence routine **405** will parse out the elements of the exercise description block and display them on the LCD. If the exercise parameter is 0, its display may be omitted. The display sequence is as follows:

STRETCH Exercises:
Name of Exercise
delay 3 seconds
Repetitions Setting (if not 0)
delay 3 seconds
Sets Setting (if not 0)
delay 3 seconds

The sequence is repeated three times **406** after which the microprocessor will enter its low power sleep mode **407** until a keypad key is activated and the resultant activation interrupt is detected. The sequence will then restart at the top.

FIG. **18** Shows a flow diagram of the Version-B DTA firmware stopwatch START/STOP and LAP/RESET key activation sequences

If the START/STOP key is detected **408**, the program will first determine if the timer is already running **409**. If so, the timer is stopped **410** and the program will wait for another key activation **411**. If the next detected key activation is a function key, the program returns to the point of interruption **415**. If the next detected key activation is a numeric key **412**, the key is ignored. If the next key detected is the LAP/RESET key, the timer is reset to its initial value **416** depending if it was in count down or count up mode. If the next key detected is the START/STOP, the timer is re-enabled **413** and the timer continues.

If the timer was not running initially when the START/STOP key is detected, the "Set Minutes" prompt **414** is displayed. The program waits for another key activation **417**. If the START/STOP key is activated, the timer goes into its count up mode **418**. If a digit is pressed, the program will accept a minutes setting **419**. Any other key will be ignored.

Following Inputting minutes, the program continue at Marker B for ENTER key processing **420**. Upon completion of entering a valid minutes setting, the program will prompt for seconds input **421** & **422**. The program continues at Marker B for ENTER key processing **423**. Upon completion of entering a valid seconds setting, the program waits for another key activation **424**. If the LAP/RESET key is detected, the program resets the timer function and redisplays the "Set Minutes" prompt **414**. If the START/STOP key is detected, the count-down timer mode is started **425** and the program displays the elapsed time while waiting for time-out or another key activation **429**.

If the LAP/RESET key is activated, the program will disable the display function while maintaining the timer **428**. The program will wait for the LAP/RESET key activated again **427**. When detected, the display function is re-enabled **426**.

If the START/STOP key is activated, the timer is stopped **430** and processing continues at Marker F **431**.

Any other key will be ignored. The program will monitor the count in count-down mode looking for the last 5-seconds remaining **432**. When detected, the alarm will beep on each second **433**. When the timer expires **434**, the alarm will sound several beeps and the timer is stopped **435**. Processing then continues at Marker F **436**.

FIG. **19** Shows a flow diagram of the Version-B DTA firmware PLATES/LBS, REPS., SETS, DISTANCE, DURATION, SPEED, INCLINE, LEVEL, and CALORIES keypad key activation sequences

On the Version-B DTA, each function has an individual key although the actual signal to the microprocessor may be shared. The context in which it is activated will distinguish the desired function.

If the PLATES/LBS key is detected **437**, the program checks to see if the EDB is Anaerobic **438**, if not then a special error message is displayed and the program resumes at the point of interruption **439**. If so, the program will extract the current plates/lbs value and display it **440**. The program will now accept a new value **441** and then continue at Marker B for ENTER key processing **442**. Upon completing a valid input, the program resumes at the point of interruption **442**.

If the REPS. key is detected **458**, the program checks to see if the EDB is Aerobic **459**, if so then a special error message is displayed and the program resumes at the point of interruption **460**. If not, the program will extract the current repetitions value and display it **461**. The program will now accept a new value **462** and then continue at Marker B for ENTER key processing **443**. Upon completing a valid input, the program resumes at the point of interruption **464**.

If the SETS key is detected **479**, the program checks to see if the EDB is Aerobic **480**, if so then a special error message is displayed and the program resumes at the point of interruption **481**. If not, the program will extract the current sets value and display it **482**. The program will now accept a new value **483** and then continue at Marker B for ENTER key processing **484**. Upon completing a valid input, the program resumes at the point of interruption **485**.

If the SPEED. key is detected **444**, the program checks to see if the EDB is Aerobic **445**, if not then a special error message is displayed and the program resumes at the point of interruption **446**. If so, the program will extract the current speed value and display it **447**. The program will now accept a new value **448** and then continue at Marker B for ENTER key processing **449**. Upon completing a valid input, the program resumes at the point of interruption **450**.

If the DISTANCE key is detected **465**, the program checks to see if the EDB is Aerobic **466**, if not then a special error message is displayed and the program resumes at the point of interruption **467**. If so, the program will extract the current speed value and display it **468**. The program will now accept a new value **469** and then continue processing at Marker B **470**. Upon completing a valid input, the program resumes at the point of interruption **471**.

If the INCLINE key is detected **486**, the program checks to see if the EDB is Aerobic **487**, if not then a special error message is displayed and the program resumes at the point of interruption **488**. If so, the program will extract the current incline value and display it **489**. The program will now accept a new value **490** and then continue at Marker B for ENTER key processing **491**. Upon completing a valid input, the program resumes at the point of interruption **492**.

If the LEVEL key is detected **451**, the program checks to see if the EDB is Aerobic **452**, if not then a special error message is displayed and the program resumes at the point of interruption **453**. If so, the program will extract the current level value and display it **454**. The program will now accept a new value **455** and then continue at Marker B for ENTER key processing **456**. Upon completing a valid input, the program resumes at the point of interruption **457**.

If the CALORIES key is detected **472**, the program checks to see if the EDB is Aerobic **473**, if not then a special error message is displayed and the program resumes at the point of interruption **474**. If so, the program will extract the current calories value and display it **475**. The program will now accept a new value **476** and then continue at Marker B for ENTER key processing **477**. Upon completing a valid input, the program resumes at the point of interruption **478**.

If the DURATION key is detected **493**, the program will extract the current duration value and display it **494**. The program will now accept a new value **495** and then continue at Marker B for ENTER key processing **496**. Upon completing a valid input, the program resumes at the point of interruption **497**.

F. Application Software Description

The application software, henceforth referred to as Physicalc, is described below with reference to the drawings specified. Sample screens from the program are included for clarity.

FIG. **20** Shows a connectivity diagram for the application software program depicting I/O screens and database file relationships.

Following is a function by function description of the Physicalc software including a representative I/O screens and descriptions of the related database files associated with the function.

The main menu for the Physicalc software program is composed of nine graphical icons **500**, **501**, **502**, **503**, **504**, **505**, **506**, **507**, & **508**. By selecting the icon using a cursor pointing device (i.e. mouse) or keyboard (Tab and Enter keys), the underlying function is accessed. An associated menu bar is also available from which the user can select the function.

When the user selects the Administration icon **506**, the Administrative Functions capabilities of the software are accessed. A graphical interface screen **509** is presented to the user. See FIG. **25** for screen representation.

By selecting a function presented on the screen or through selection from an associated menu bar, the user can perform one of the following tasks:

1. Examine pertinent file statistics.
2. Set the system access security for all users.
3. Chart the facility's exercise equipment usage.
4. Pack (remove deleted records) the database files.
5. Set system wide preferences.
6. Inventory DTA units.
7. Create mailing lists and labels for the clients stored in the system.

Connected to this function are two database files. The first file is called prefs.dbf **510** and is a small database file used to hold system wide preference parameters. A description of a data stored in the preference table includes the following:

1. A list of membership type categories.
2. A list of Trainer classification types.
3. The hours of operation for the facility.
4. The host PC communications port the DTA Programming Stand is connected to.

The other database file is called dta.dbf **511**. This will hold the status of Digital Training Assistant units programmed by Physicalc during the current session. It is used in the Inventory DTA function in both the Administrative functions and Reports & Graphs functions of the program. A description of the records included in the database includes:

1. The client's membership number.
2. The date the DTA was uploaded.
3. The date the DTA was downloaded.

When the user selects the Configure Facility icon **505**, the Facility Configuration function of the software is activated. A graphical interface screen **512** is used to input and display data records from the facility configuration database. Refer to FIG. **26** for screen representation.

This function is used to create a database of exercises and exercise equipment that are available at the facility in which the system is installed. An additional capability is the ability to create a series of template files containing a subset of

exercises from the facility's master database. These templates can then be used to quickly construct exercise routines with pre-defined equipment lists.

Connected to this function are four database files. The facility configuration database file called facility.dbf **515** will contain records representing all the exercise equipment available at the facility where the Physicalc program is to be used. The user will use the Facility Exercise Configuration screen and associated menu bar to build the file's records. A complete description of an exercise includes the following:

1. Muscle Group being worked
2. Classification (free weights, nautilus machine, etc.)
3. Exercise Group (Aerobic, Anaerobic, or Stretch)
4. Exercise Name (50 characters maximum)
5. Name as Displayed by DTA (16 characters maximum)
6. Lookup Table # (if name comes from the hardcoded lookup table)

Exercises that are specified as AEROBIC will have the additional following information:

1. Up to seven (7) Mechanical settings for equipment setup (specified as a letter or number)
2. Duration (specified as either seconds, minutes, or hours)
3. Level Setting (either a letter or number)
4. Speed Setting (specified as either mph, kph, fpm, or mpm)
5. Calories Setting
6. Distance Setting (specified as either kilometers, meters, miles, or feet)
7. Incline (specified as degrees, used on treadmills)
8. Intensity Setting (target pulse rate)

Exercises that are specified as ANAEROBIC will have the additional following information:

1. Up to seven (7) Mechanical settings for equipment setup (specified as a letter or number)
2. Pounds/Plates Setting (pounds for free weights and plates for weight machines)
3. Repetitions setting
4. Sets setting (number of groups of repetitions)
5. Duration (specified as either seconds, minutes, or hours)

Exercises that are specified as STRETCH will have the additional following information:

1. Repetitions setting
2. Sets setting (number of groups of repetitions)
3. Duration (specified as either seconds, minutes, or hours)

A supplemental read-only table called lookups.dbf **516**, contains a list of exercise names and associated code numbers. This table mimics the table of hard-coded exercise names found in the Digital Training Assistant (DTA) hand-held computer's firmware program. It will be used to substitute a number in place of a 16 character exercise name when programming the DTA units. This saves considerable memory space in the DTA.

As mentioned earlier, as a record is built, it may be added to Exercise Template files **514** with the unique naming convention:

xxxxxtpl.dbf (where xxxxx=first 5 characters specified by user)

The user may choose to add the current completed exercise record to a specified template table file using the Add to Template screen push button or Template→Add to Template menu option The end result of adding to the template is a custom list of exercises that the user can then copy to a client's workout routine as a skeleton to build upon. A Template file shares the same structure as the facility.dbf database file.

The Muscle Group Files **513** are composed of two read-only database files. The genmus.dbf database file is a read-only list of muscle groups arranged by major body part then alphabetically by major muscle group in that part of the body. The specmus.dbf database file is a read-only list of muscle groups arranged by major body part then alphabetically by specific muscle group in that part of the body

When the Client Setup icon **507** is selected, the client Information Setup function is activated. This function allows the user to input and display demographic and medical data into a client database file. Activation of the function presents the Client Information Screen **517**. Refer to FIG. **27** for screen representation.

This screen is one of two used to input and display the demographic, membership and medical data elements of the selected client database. There are controls for moving around in the database as well as search, remove, and adding capabilities. Additional controls provide access to the Medical Info Screen as well as accessing the Workout Builder utility.

Connected to this screen is a single database file called client.dbf **518**. This database will contain records representing demographic and medical information from a set of clients utilizing the Physicalc system. The user will use the Client Info Screen **517**, Medical Info Screen **519** and associated menus to build records. A complete description of a client database record includes the following:

Personal Data:
1. Full Name
2. Date of Birth
3. Age (computed to current date)
4. Sex

Home Address:
1. Street Address
2. Apartment/Unit Number
3. City
4. State
5. ZIP code (w/4-digit extension)

Business Information:
1. Business Name
2. Job Title
3. Street Address
4. Department/MS
5. City
6. State
7. ZIP code (w/4-digit extension)

Membership Data:
1. Membership #
2. Membership Type
3. Expiration Date
4. Personal Trainer's Name

Phone Numbers:
1. Home Phone Number
2. Business Phone Number

Relevant Notes

Access to the Medical Information Screen **519** is through the Client Information Screen Medical push-button or menu selection. Refer to FIG. **28** for screen representation.

This screen is used to input and display medical oriented data elements of the client database file. The following client.dbf database file record data items are accessed through the Medical Information screen:

Cardiac Risk Factors:
1. Age (computed to the current date)
2. Name of pulmonary condition (if any)
3. Existence of a heart condition
4. Existence of a heart conditions in the family
5. Existence of a Hypertension
6. Level of daily stress (low, medium, or high)
7. Smoking habits (how much and for how long)
8. Maximum Stress Test score
9. Physical activity level
10. Cholesterol HDL & LDL values

Other Medical Information:
1. Existence of Asthma
2. Existence of any Hernia
3. Use of contact lenses or Glasses
4. Existence of Diabetes
5. History of being frequently tired
6. Existence of Allergies
7. Current Pregnancy
8. Family history of problem pregnancies
9. Alcohol Intake (drinks per week)
10. Number of common colds per year
11. Caffeine intake (cups or can per day)
12. List of major surgeries
13. List of current medications
14. Date of last physical (by physician)
15. Physician's name
16. Physician's phone number (or other in case of emergency number)

Orthopedic Conditions:
1. client's height
2. client's weight
3. Existence of Cervical problems
4. Existence of Thoracic problems
5. Existence of Lumbar problems
6. Existence of Sciatica problems
7. Existence of Shoulder problems
8. Existence of Elbow problems
9. Existence of Wrist problems
10. Existence of Finger problems
11. Existence of Sacroiliac problems
12. Existence of Hip problems
13. Existence of Knee problems
14. Existence of Ankle problems
15. Existence of Toe problems
16. Description of injuries

Each client entered into the system may have an associated fitness workout program designed for him/her. Connected to this screen are four database files. The Workout Builder function of the software is accessed from the Client Information Screen Workout push-button or menu selection. The Client Workout Profile Builder screen **520**, is presented to assist the user in constructing a workout program for the currently selected client record in the client.dbf database file. Refer to FIG. **29** for screen representation.

The workout.dbf database file **521** is used as a template for creating individual Client Workout Routine **522** and Client Workout History **523** database files. In creating the individual client's workout routines, the structure from workout.dbf is copied to create the new Client Workout Routine file. A complete description of a workout routine includes the following:

1. Exercise Group (Aerobic, Anaerobic, or Stretch)
2. Machine/Exercise Mechanical Settings
3. Parametric Exercise Settings (not necessary all listed used per exercise)
   a. Pounds/Plates
   b. Repetitions
   c. Sets
   d. Duration
   e. Distance
   f. Speed
   g. Intensity
   h. Incline
   i. Calories
   j. Level
4. Exercise Philosophy
   a. When to increase parameter
   b. What parameter to increase
   c. By how much to increase parameter
   d. How to continue the increase parameter philosophy
   e. When to decrease parameter
   f. What parameter to decrease
   g. By how much to decrease parameter
   h. How to continue the decrease parameter philosophy
5. Routine Time Stamping
   a. Start of Routine Date
   b. Date of latest Completion
   c. Enable Days
   d. Set Number Designation

The client Workout Routine file is uniquely named using the following formula:

xxxxyyyy.dbf

xxxx: first 4-letters of th client's last name taken from the currently selected record in the Client Database file last name (last_name) field. If the client's last name is shorter than four letters, the lower case letter 'x' is appended until four letters are created.

yyyy: last 4-digits of client's membership number taken from the currently selected record in the Client Database membership number (member_no) field. If the client's membership number is shorter than four digits, the number 9 is appended until four digits are created.

The Client Workout Routine database files **522**, henceforth referred to as CWR, contains a single copy of the client's computed workout routine for the day it was requested. It will be used to create a new workout routine based upon past performance and special "philosophy" parameters entered using the Exercise Philosophy Configuration screen **524**. The routine is then scheduled over a 7 day period using the Schedule Routine Screen **525**.

The client may have up to 7 different workout routines designated as sets. The CWR file is grouped by set first and then sorted by exercise in the order they were input into the system.

The Client Workout History database files **523**, henceforth referred to as CWH, are uniquely named using the follow formula:

xxxxyyyy.his

xxxx: first 4-letters of the client's last name taken from the currently selected record in the Client Database file last name (last name) field. If the client's last name is shorter than four letters, the lower case letter 'x' is appended until four letters are created.

yyyy: last 4-digits of client's membership number taken from the currently selected record in the Client Database membership number (member_no) field. If the client's

membership number is shorter than four digits, the number 9 is appended until four digits are created.

The CWH contain a history of completed exercises. This file is composed of records representing exercises and associated performance parameters. Each record contains a date stamp allowing the Report & Graphs function to generate reports and graphs of client performance over a selected period of time. Following a workout session, the client's executed workout information data package is transferred from the DTA, modified (irrelevant fields removed, i.e. mechanical settings), date stamped, and appended to the client's CWH file. A complete description of a data stored in the records of a CWH file includes the following:

1. Full Exercise Name
2. DTA Exercise Name
3. Exercise Group (Aerobic, Anaerobic, or Stretch)
4. Parametric Exercise Settings (not necessary all listed used per exercise)
   a. Pounds/Plates
   b. Repetitions
   c. Sets
   d. Duration
   e. Distance
   f. Speed
   g. Intensity
   h. Incline
   i. Calories
   j. Level
5. Client's Weight
6. Client's Pulse
7. Client's Blood Pressure
8. Date of routine completion
9. Completed as Prescribed Flag

The trainer supplements an exercise's description by adding a "Philosophy" to the exercise. The Philosophy describes the action to be performed on selected exercise parameters when the client performs (or not) the exercise in the specified fashion. Pushing the Set Philosophy screen push button activates the Exercise Philosophy Configuration screen **524** where the trainer can now assign his philosophy parameters to the exercise. Refer to FIG. **30** for screen representation.

The exercise can have two of its parameters increase and two of its parameters decrease automatically. Alternatively, the exercise can choose to automatically remove itself when a specific condition is met. One action for each function is designated as the primary action and the one action is the secondary action. Both functions require a "trigger event" to occur before the increment, decrement or removal can occur. This trigger event involves checking a selected parameter against a selected value using a logical type operator. When the trigger event evaluates as "True", the primary action occurs. A limit value is included to prevent the primary action from setting a parameter too high or too low. After the trigger event occurs, it is automatically reset with the new primary action parameter value as the base.

An optional 2nd Action may also occur when the trigger event occurs. A separate check box for both the Increase and Decrease functions is provided. A second set of parameter popup lists is used to set a secondary action. The secondary action will occur every time the trigger event occurs until its own selectable limit value is reached of the primary action's limit is reached, which ever comes first.

NOTE: Program code will prevent the user from setting the primary and secondary action to the same parameter

The user specifies the parameter to be automatically modified. Next he chooses the numeric value to either add or subtract from the parameter. The user can chose a discrete value or a percentage of the current value with the % radio button. Next he selects a modification limit for the parameter. When the limit is attained, further modification is disabled. Next the trigger event is decided by selecting a different parameter or the additional options of elapsed days or completed as prescribed. Then the trigger parameter operator is selected from one of the following:

= Equal to

< Less than

<= Less than or equal to

> Greater than

>= Greater than or equal to != Not equal to

The comparison value is set next. The trigger event will be evaluated as either True or False. A True evaluation constitutes a trigger event. The user may for either case (increase or decrease) specify a second action to be performed. Checking the Trigger 2nd Action checkbox will enable a second different parameter to be augmented. The user must again select the parameter, the increment or decrement value or percentage and the limit for the parameter.

Every time Physicalc is requested to create a client's daily workout routine, the trigger event for each exercise record in the client's CWR **522** is check against the most recent corresponding exercise record stored in the client's CWH **523** file. If the trigger conditions are determined as being met, the action parameters in the CWR **522** record is modified according to the action specification. For the initial workout, the date stored in a special field is used to determine any elapsed days.

The primary and secondary actions may also be incremented or decremented as a percentage of the current value. Selection of the % radio button will activate this feature.

A removal of an exercise as the primary action is a single shot event and once triggered, it remains disabled until explicitly set by a new philosophy setting.

Pushing the Set Schedule screen push button activates the Schedule Routine screen **525** where the trainer can now schedule exercise routines. Refer to FIG. **31** for screen representation.

Exercises can be grouped into 1 of 7 sets. This allows for the formulation of exercise set rotations found in many professional and competitive training regiments. A field in the records of the CWR database file called ex_set designates, the set it belongs to. A further enhancement of this function is the ability to schedule these sets over an arbitrary 7-day period. The period is designated as DAY #1 through DAY #7 with DAY #1 being set to a specific day of the week (stored in its own field). You can now schedule any set to any day using the Schedule Routine Screen.

The trainer simply types a number under the Day X field to designate that on that day, the exercises with the matching set number are used to formulate the daily workout routine. The software calculates the day number using the current date and the value stored in the special field.

Program code for this screen will assign a bit in a byte for each day of the week (bit 1=Day 1, etc.). When the Done screen push button is pressed, the program will look at the exercise set field of ALL records in the selected CWR database file. It will set the bit in the record's field called week_sched corresponding to each day the exercise set was found in.

For example, an exercise record designated as belonging to Set #1 with the following Schedule Routine Screen settings will produce the resulting byte:

DAY 1 = Set 1 DAY 2 = Set 2 DAY 3 = Set 1 DAY 4 = Set 3
DAY 5 = Set 1 DAY 6 = Set 6 Day 7 = Set 2

	7	6	5	4	3	2	1	0
week_sched	0	0	1	0	1	0	1	X

The philosophy day counting rules count total elapsed days including those not designated for the particular set. The When parameter must account for this when setting Elapsed Days.

If the user selects the Automatic Mode icon **508**, the Automatic Mode function is executed. Refer to FIG. **32** for screen representation.

The Automatic Mode screen **526**, is used to operate all the functions of the utility. When the client's membership card is read by the bar-code or magnetic strip reader it will present the membership number (10-digit max) to the Automatic Mode program code. The number will then be passed to the Program DTA Procedure and used to access the client's CWR **528** and CWH **527** to formulate the workout routine for the day. The workout routine is formatted into the upload exercise routine data package and transmitted to the DTA. Status of the operation is reflected in the Status field.

The user can also elect to receive data from a DTA (download). The function sends the appropriate command to the DTA to initiate downloading of its data. The data is parsed and reformatted to be incorporated back into the client's CWH file. Additional functionality allows the user to instantly perform some limited graphing functions for the client.

If the user selects the Blood Pressure icon **501**, the Blood Pressure Input mode function is executed. Refer to FIG. **33** for screen representation.

The Blood Pressure Input screen **530**, is used to operate all the functions of the utility. When the client's membership card is read by the bar-code or strip reader it will present the membership number (10-digit max) to the Blood Pressure Input program code. The number will then be used to access the client's CWH **531** to allow fast input of the blood pressure systolic and diastolic readings to their appropriate fields in the database's records without having to manually open the file and insert the data.

When the Trainers Setup icon **503** is selected, the Trainer Information Setup function is activated. This function allows the user to input and display demographic and employment data into the trainer database file **533**. Activation of the function presents the Trainer Information Screen **532**. Refer to FIG. **34** for screen representation.

This screen is used to input and display the demographic and employment data elements of the trainer database file **533**. There are controls for moving around in the database as well as search, remove, and adding capabilities.

Connected to this screen is a single database file called trainer.dbf **533**. This database will contain records representing demographic and employment information for a set of trainers utilizing the Physicalc system. The user will use the Trainer Info Screen **532**, and associated menu to build records. A complete description of a Trainer database record includes the following:

Personal Data:
1. Full Name
2. Date of Birth
3. Age (computed to current date)
4. Sex

Home Address:
1. Street Address
2. Apartment/Unit Number
3. City
4. State
5. Zip Code (w/4-digit extension)

Employment Data:
1. Employee #
2. Employee Type
3. Start Date
4. Security Clearance Level
5. Security Password
6. Work Schedule

Phone Numbers
1. Home Phone Number

Relevant Notes

Selecting the Reports & Graphs icon **504** will activate the Report and Graph generating function. This function will gather data elements from selected records in the DTA usage file **535**, the client database riles **536**, and the Client Workout History database files **537**. A single main screen is used to operate the function. Refer to FIG. **35** for screen representation. Through the screen's graphical icons, the following reports and graphs can be generated:

1. Graphing a client's weight over a specified time period.
2. Graphing a client's pulse over a specified time period.
3. Graphing a client's blood pressure over a specified time period.
4. Graphing a selected exercise's parameter for a specific client over a specified time period.
5. Create mailing lists and produce mailing labels for specified clients.
6. Report a client's workout frequency.
7. Report a selected client's demographic and medical history.
8. Report on equipment usage over time by all clients.
9. Report Digital Training Assistant usage.

Selecting the Help icon **500** will bring up the on-line help utility allowing the user to access text describing the functions and operations of the system. A context type help is available using the F1 key when the cursor is in a particular field of a screen. A topical help is also available using the Help icon **500** or the Help menu option.

The user with appropriate access clearance can also modify the text of the help screens to instruct the users on club practices and procedures.

Selecting the Exit icon **502** will exit the program back to the Windows™ program manager.

G. Digital Training Assistant Workout Routine Program Creation Description

Central to the Physical application software are the software algorithms that create and decode exercise routine data packages. Collective they are referred to as the Program DTA procedure. Wherever necessary, the actual database file record field name is used for clarity.

The Program DTA procedure is divided into two sections. The first section involves compiling data from the Physicalc system's database files and transmitting it to a DTA (Transmit to DTA Subfunction). The second section involves the reverse process of taking data from a DTA and transferring it back into the system's database files (Receive From DTA Subfunction).

The following sections assume that the selected client's record in the currently selected Client Database file, the client's CWR, and CWH files are opened.

Transmit to DTA Subfunction

The DTA requires an upload of an exercise routine data package before use. The uploaded data consists of the exercises to be performed and the associated setup and performance parameters as determined by the trainer and the Physicalc program.

DTA Data Package Format

NOTE: All following data byte values are in their hexadecimal format as indicates by the leading 0x prefix to the number.

The DTA accepts a "data package" made up of embedded codes describing the client and the workout routine. The Program DTA Procedure will first construct the Header Block followed by the Exercise Description Blocks (EDBs) and terminated by a Footer code byte and Checksum Byte.

Header Block

The Header Block consists of 19 bytes. Any data not explicitly specified will be set to 0x0.

Byte 1	HEADER COMMAND
Byte 2	MEMBER # LSB
Byte 3	MEMBER #
Byte 4	MEMBER #
Byte 5	MEMBER # MSB
Byte 6	# OF CHARACTERS IN NAME
Byte 7	222 11111
Byte 8	4 3333 22
Byte 9	55555 4444
Byte 10	77 66666 5
Byte 11	88888 777
Byte 12	AAA 99999
Byte 13	C BBBBB AA
Byte 14	DDDD CCCC
Byte 15	FF EEEEE D
Byte 16	GGGGG FFF
Byte 17	WEIGHT LSB
Byte 18	WEIGHT MSB
Byte 19	PULSE

Header Command (Byte 1)

The header command is a byte that tells the DTA to operate in Normal Mode or Learn Mode. In Learn Mode, the DTA user can change the mechanical/ergonomic settings. In Normal Mode these changes are locked out.

Normal Mode and Learn Mode are selectable in the Workout Builder menu DTA→Program DTA submenu options.

Normal Mode=0x00

Learn Mode=0x02

Membership Number Bytes (Bytes 2–5)

The member_no field of the client's client database (client.dbt) record in the currently selected client database file contains a 10-digit maximum integer number. This number is stored as four bytes in the header block with leading 0's as required. The Least Significant Byte (LSB) is stored in the Byte 2 location and the Most Significant Byte (MSB) in the Byte 5 location.

Client's Last Name Number of Characters Byte (Byte 6)

The client's last name is loaded into the DTA up to a maximum of 16 characters. The ASCII characters that makeup the last name are packed into the next 10 bytes to conserve memory in the DTA. The Number of Characters byte will tell the DTA software how many characters to extract from it's memory to display the user's last name. The number of actual characters is determined by counting characters in the last_name field of client's record in the currently selected client database file.

Client's Packed Last Name Bytes (Bytes 7–16)

As mentioned in the above section, the ASCII characters of the client's last name are packed into 10 bytes. The ASCII value of the character is trimmed to only 5 bits, the most significant three (3) bits will be added by the DTA firmware. This, again, is done to conserve space in the limited DTA memory.

The characters are represented as the numbers 1–9 and letters A–G in the following packing diagram. In the table below, the bits of the ASCII byte representing the first letter of the client's last name (all leading and trailing blanks removed) is represented by the number 1, the second character's bits by 2, and the last by G. Bits are ordered from right to left, least significant to most significant. The program code will reformat the client's last name character string taken from the last_name field of the client's record in the currently selected client database file.

Byte 7	222 11111
Byte 8	4 3333 22
Byte 9	55555 4444
Byte 10	77 66666 5
Byte 11	88888 777
Byte 12	AAA 99999
Byte 13	C BBBBB AA
Byte 14	DDDD CCCC
Byte 15	FF EEEEE D
Byte 16	GGGGG FFF

NOTE: Any blank bytes or left over bits in the byte are set to 0

Client's Weight Bytes (Bytes 17–18)

The client's weight is stored in the CWH file's weight field. The program finds the most current record (looking at the ex_date field) and extracts the value of the weight field in the record. If no current record found then the value defaults to 0x0000.

The field is specified as three digits and a tenths of a pound digit. The program must reformat the data as follows. If there is a tenths digit other than 0 specified (i.e. 195.8 lbs. or 97.1 lbs.) the number is multiplied by 10 to remove the fraction and stored in two (2) bytes. The MSB byte's most significant bit must then be set as a flag to notify the DTA software to divide the number by 10 before displaying.

For Example:

weight field = 193.7

1) Multiply by 10 1937 = 0x791
2) Store in two bytes

	7	6	5	4	3	2	1	0
MSB	0	0	0	0	0	1	1	1
LSB	1	0	0	1	0	0	0	1

3) Set the MSB's most significant bit

	7	6	5	4	3	2	1	0
MSB	1	0	0	0	0	1	1	1

Result = 0x8791
weight field = 125.0

1) Multiply by 10 1250 = 0x4E2
2) Store in two bytes

	7	6	5	4	3	2	1	0
MSB	0	0	0	0	0	1	0	0

-continued

	7	6	5	4	3	2	1	0
LSB	1	1	1	0	0	0	1	0

Result = 0x4E2

Client's Pulse (Byte 19)

The client's pulse is stored in the CWH file's pulse field. The program finds the most current record (looking at the ex_date field) and extracts the value of the pulse field in the record. If no current record found then the value defaults to 0x00. The pulse value is stored in Byte 19 of the Header Block.

Exercise Description Blocks (EDB)

Following the creation of the Header Block, the Program DTA Procedure—Transmit to DTA Subfunction will now construct EDBs incorporating Physicalc's exercise philosophy algorithms.

Exercise Philosophy Pre-Processing

The exercise routine for the specific day is extracted from the client's CWR file. Now the philosophy feature of Physicalc uses the last completed routine matching the exercise set number as derived from the processing depicted below. The process follows a step by step procedure as follows:

1. The program first looks at the current system date and determines what day of the week it is, Monday, Tuesday, etc.
2. The program then looks at the d_of_week field in the first record of the CWR file. By design, all records in the CWR will have the same value in the d_of_week field.
3. Using the d_of_week value the program computes what day of the week number it is for the particular client.
4. The program will now, starting at the first record, scan through each record in the CWR and look at the record's week_schd field byte. If a bit is set in the computed Day # bit position AND the ex_enable field is logical True, the exercise record is copied to a temporary cursor table (henceforth called Table A).
   NOTE: The program keeps track of where the record came from in the original CWR table. The modified records in Table A will eventually be copied back into their original positions in the CWR file.
   The resulting cursor table (Table A) contains all exercises and associated parameters that are supposed to be performed on this workout Day #
5. The ex_set field value from the first record in Table A is stored to a temporary variable named m.ex_set.
   NOTE: By design, all records for a specific Day # will belong to the same Exercise Set #. You can only schedule one Set # in any Day #.
6. The program now looks at the CWH file and starting at the last record and searching up to the top, the program compares the m.ex_set variable to each CWH record's ex_set field value.
   When the first match is made, the record number is noted (stored in a temporary variable, sendrec) The search continues (going up towards the first record) until the first record found with a mismatch between m.ex_set and ex_set field value. The record number +1 is noted (stored in a temporary variable, startrec)
   NOTE: If no matches are found then this is the first time the routine (Set) is executed and the Table A records are used as is. Proceed to Step #35
   We now have the starting and ending record numbers (sendrec & startrec) for the latest completed exercise routine history matching the exercise Set number to be performed today. The records are checked for blank exercise field indicating blood pressure only records. Those records are skipped and the others are copied in ascending record order (normal) to a temporary cursor table opened in the next available word area (henceforth called Table B).
7. The program will now extract data from the following Table A's (currently pointed to) record fields and place the values into temporary variables

Table	Field	Variable
Table A	inc_type	m.inc_type
Table A	inc_what	m.inc_what
Table A	inc_by	m.inc_by
Table A	inc_per	m.inc_per
Table A	inc_until	m.inc_until
Table A	inc_when	m.inc_when
Table A	inc_op	m.inc_op
Table A	inc_wby	m.inc_wby
Table A	inc_trig	m.inc_trig
Table A	inc_2what	m.inc_2what
Table A	inc_2by	m.inc_2by
Table A	inc_2per	m.inc_2per
Table A	inc_2until	m.inc_2until
Table A	dec_type	m.dec_type
Table A	dec_what	m.dec_what
Table A	dec_by	m.dec_by
Table A	dec_per	m.dec_per
Table A	dec_until	m.dec_until
Table A	dec_when	m.dec_when
Table A	dec_op	m.dec_op
Table A	dec_wby	m.dec_wby
Table A	dec_trig	m.dec_trig
Table A	dec_2what	m.dec_2what
Table A	dec_2by	m.dec_2by
Table A	dec_2per	m.dec_2per
Table A	dec_2until	m.dec_2until
Table A	ex_set	m.ex_set

Now examine the Increase Parameter philosophy as follows

8. The program then examines the m.inc_type variable and if=3, no action is performed and the program advances to Step #21. If other than 3, the processing continues
9. The next action is to examine the parameter name stored in the m.inc_when variable. The following table equates the numeric code in the m.inc_when variable with the Table B field to examine or special action to perform:

Numeric Code	Test Parameter	Table B field or Action
1	Duration	duration
2	Distance	distance
3	Speed	speed
4	Intensity	intensity
5	Incline	incline
6	Calories	calories
7	Level	level
8	Elasped Days[1]	Compute Elasped Days
9	Completed[2]	Compute Completed

[1]The Elasped Days test parameter is computed by looking at the ex_date field in the Table B record and subtracting from the current date.

-continued

Numeric Code	Test Parameter	Table B field or Action

[2]The Completed test parameter is computed by examining the completed field in the Table B's record. If the field contains a logical True, then the program will search backwards in the CWH (for a date before the current ex_date value in Table B) for a record matching the exercise and ex_set fields from Table A. The program will count how many consecutive record matches have a logical True in their completed filed. The counting stops as soon as a matching record has a logical False in the field. This number then represents the number of consecutively completed (without modification) prescribed exercises and becomes the completed count. If the current record completed field in Table B is logical false, the completed count is set to 0.

The data from the indicated Table B field or Action is stored in the temporary variable, lookat.

10. The program now examines the operator action stored in the m.inc_op variable. Using the decoded action:

Symbol	Code #	Function
>	1	Greater Than
<	2	Less Than
=	3	Equal To
>=	4	Greater or Equal To
<=	5	Less or Equal To
!=	6	Not Equal

The program evaluates the operand variable lookat and the test value in m.inc_wby.

NOTE: The numeric value in m.inc_wby can be used directly without moving to another variable.

11. If the evaluation is false (i.e. 100>120) The program aborts the current processing and continues at Step #21. If the evaluation is True (i.e. 120>100) the processing continues.

Have now established an active Increase trigger condition

12. If the value of m.inc_type=2 then the ex_enable field of the TABLE A's record is set to logical false. The program jumps to Step #21.

If the value of m.inc_type=1 the program will extract the data stored in the Table A field named by the m.inc_what variable and store the data it in a temporary variable, incoperand.

13. The logical value of the m.inc_per variable determines if we increase by a percentage or discrete value.

If it is logical True, the program computes what m.inc_ by percent of incoperand is and then adds that result to incoperand storing the sum back into incoperand.

NOTE: The value in m.inc_by is a number and can be used directly from the variable

If m.inc_per is logical False, the value of m.inc_by is added directly to incoperand and the sum is put back into incoperand.

14. We now do a limit check by comparing incoperand against the numeric value that was stored in the m.inc_ until variable.

If incoperand exceeds m.inc_until, the current value of incoperand is changed to the value of m.inc_until.

If incoperand is less than m.inc_until, then no change.

15. We now check to see if a 2nd Action is to be performed. The program looks at the logical value stored in the m.inc_trig variable.

If logical false, the program continues at Step #20.

If true, the process continues

16. The program will extract the data stored in the Table A field named in the m.inc_2what variable and store the data it in a temporary variable incoperand2.

17. The logical value of the m.inc_2per variable determines if we increase by a percentage or discrete value.

If it is logical True, the program computes what m.inc_ 2by percent of incoperand2 is and then adds that result to incoperand2 storing the sum back into incoperand2.

NOTE: The value in m.inc_2by is a number and can be used directly from the variable

If m.inc_2per is logical False, the value of m.inc_2by is added directly to incoperand2 and the sum is put back into incoperand2.

18. We now do a limit check by comparing incoperand2 against the numeric value that was stored in the m.inc_ 2until variable.

If incoperand2 exceeds m.inc_2until, the current value of incoperand2 is changed to the value of m.inc_ until2.

If incoperand2 is less than m.inc_2until, then no change

The new parameters have now been computed and they are copied back to the fields in the current record of the Table A.

19. The incoperand2 value is stored back to Table A's field named in the m.inc_2what variable

20. The incoperand value is stored back to Table A's field named in the m.inc_what variable

Now examine the Decrease Parameter philosophy as follows

21. The program then examines the m.dec_type variable and if=3, no action is performed and the program advances to Step #34. If other than 3, the processing continues

22. The next action is to examine the parameter name stored in the m.dec_when variable. The following table equates the numeric code in the m.dec_when variable with the field Table B field to examine or special action to perform:

Numeric Code	Test Parameter	Table B field or Action
1	Duration	duration
2	Distance	distance
3	Speed	speed
4	Intensity	intensity
5	Incline	incline
6	Calories	calories
7	Level	level
8	Elasped Days[1]	Compute Elasped Days
9	Completed[2]	Compute Completed

[1]The Elasped Days test parameter is computed by looking at the ex_date field in the Table B record and subtracting from the current date.

[2]The Completed test parameter is computed by examining the completed field in the Table B's record. If the field contains a logical True, then the program will search backwards in the CWH (for a date before the current ex_date value in Table B) for a record matching the exercise and ex_set fields from Table A. The program will count how many consecutive record matches have a logical True in their completed filed. The counting stops as soon as a matching record has a logical False in the field. This number then represents the number of consecutively completed (without modification) prescribed exercises and becomes the completed count. If the current record completed field in Table B is logical false, the completed count is set to 0.

The data from the Table B field or Action is stored in the temporary variable, lookat

23. The program now examines the operator action stored in the m.dec_op variable. Using the decoded action:

Symbol	Code #	Function
>	1	Greater Than
<	2	Less Than
=	3	Equal To
>=	4	Greater or Equal To
<=	5	Less or Equal To
!=	6	Not Equal

The program evaluates the operand lookat and the test value in m.dec_wby.

24. If the evaluation is False (i.e. 100>120) The program aborts the current processing and continues at Step #34.
If the evaluation is True (i.e. 120>100) the processing continues.

Now have established an active Decrease trigger condition

25. If the value of m.dec_type=2 then the ex_enable field of the Table A is set to logical false. The program jumps to Step #34.

If the value of m.dec_type=3 the program will extract the data stored in the CWR field named in by m.dec_what variable and store the data it in a temporary variable decoperand.

26. The logical value of the m.dec_per variable determines if we decrease by a percentage or discrete value.
If it is logical True, the program computes what m.dec_by percent of decoperand is and then subtracts that result from decoperand storing the difference back into decoperand.
NOTE: The value in m.dec_by is a number and can be used directly from the variable.
If m.dec_per is logical False, the value of m.dec_by is subtracted directly from decoperand and the difference is put back into decoperand.

27. We now do a limit check by comparing decoperand against the numeric value that was stored in the m.dec_until variable.
If decoperand is less than m.dec_until, the current value of decoperand is changed to the value of m.dec_until.
If decoperand is more than m.dec_until, then no change.

28. We now check to see if a 2nd Action is to be performed. The program looks at the logical value stored in the m.dec_trig variable.
If logical false, the program continues at Step #33
If true, the process continues

29. The program will extract the data stored in the Table A field named in the m.dec_2what variable and store the data it in a temporary variable decoperand2.

30. The logical value of the m.dec_2per variable determines if we decrease by a percentage or discrete value.
If it is logical True, the program computes what m.deinc_2by percent of decoperand2 is and then subtracts that result from decoperand2 storing the difference back into decoperand2.
NOTE: The value in m.dec_2by is a number and can be used directly from the variable
If m.dec_2per is logical False, the value of m.dec_2by is subtracted directly from decoperand2 and the difference is put back into decoperand2.

31. We now do a limit check by comparing decoperand2 against the numeric value that was stored in the m.dec_2until variable.
If decoperand2 is less than m.dec_2until, the current value of decoperand2 is changed to the value of m.dec_until2. If decoperand2 is more than m.dec_2until, then no change.

The new parameters have now been computed and they are copied back to the fields in the current record of the Table A.

32 The decoperand2 value is stored back to Table A's field named in the m.dec_2what variable

33 The decoperand value is stored back to Table A's field named in the m.dec_what variable

A single exercise record has been processed

34. The program then increments the record pointer for both Table A and Table B repeating the process from Step #7 until all records in Table A have been processed

All records in Table A have been processed

35. All of the records from Table A are now copied back to their original positions in the CWR file. Table A is still intact and will be used in the following sections.

Exercise Description Block Compiling

The Table A created above contains descriptions of exercises to be performed in the order presented in the table. The DTA units require a specific format for a complete exercise description called an Exercise Block. There are six distinct exercise block formats. They are listed as follows:

1. Aerobic Exercise with Embedded Exercise Name
2. Aerobic Exercise with Lookup Table Code Exercise Name
3. Anaerobic Exercise with Embedded Exercise Name
4. Anaerobic Exercise with Lookup Table Code Exercise Name
5. Stretch Exercise with Embedded Exercise Name
6. Stretch Exercise with Lookup Table Code Exercise Name

The Exercise Blocks are appended to the Header Block in the DTA Data Package Buffer. They follow one after the other until all the exercises have be reformatted and added to the DTA Data Package Buffer.

The ex_group field will indicate which type of exercise and the table_no field indicates if the Embedded Exercise Name is used or table lookup code number. Use the following table to determine which of the 6 exercise group classifications is represented by the Table A exercise record.

ex_group	table_no	Exercise Group
Aerobic	0	Aerobic Exercise with Embedded Exercise Name
Aerobic	Other than 0	Aerobic Exercise with Lookup Table Code Exercise Name
Anaerobic	0	Anaerobic Exercise with Embedded Exercise Name
Anaerobic	Other than 0	Anaerobic Exercise with Lookup Table Code Exercise Name
Stretch	0	Stretch Exercise with Embedded Exercise Name
Stretch	Other than 0	Stretch Exercise with Lookup Table Code Exercise Name

The following sections describe how each exercise record in Table A is re-formatted to the corresponding exercise block

Aerobic Exercise w/Embedded Exercise Name

NOTE: The following table lists the bytes in the formatted exercise block starting with the arbitrary number 1. The actual byte number in the DTA Data Package Buffer will depend upon its actual position amongst the other re-formatted exercise blocks.

BYTE 1	Exercise Group	
BYTE 2	BIT 7:	Spare
	BIT 6:	Setting 7 1 = ALPHA 0 = NUMERIC
	BIT 5:	Setting 6 1 = ALPHA 0 = NUMERIC
	BIT 4:	Setting 5 1 = ALPHA 0 = NUMERIC
	BIT 3:	Setting 4 1 = ALPHA 0 = NUMERIC
	BIT 2:	Setting 3 1 = ALPHA 0 = NUMERIC
	BIT 1:	Setting 2 1 = ALPHA 0 = NUMERIC
	BIT 0:	Setting 1 1 = ALPHA 0 = NUMERIC
BYTE 3	BITS 7-5:	# OF SETTINGS (0–7)
	BITS 4-0:	Setting #1 (0–32 or A–Z)
BYTE 4	333 22222	Setting #2 & #3
BYTE 5	5 44444 33	Setting #3, #4 & #5
BYTE 6	6666 5555	Setting #5 & #6
BYTE 7	XX 77777 6	Setting #6 & #7
	BIT 6:	Spare
	BIT 7:	Spare
BYTE 8	BITS 7-0:	INTENSITY beats per minute (0–255)
BYTE 9	BIT 7:	1 = LEVEL IS ALPHA 0=LEVEL IS NUMERIC
	BITS 6-0:	LEVEL (0–127 or A–Z)
BYTE 10	BITS 7-5:	RATE: 0=MPH 1=KPH 2=FPM 3=MPM 4=SPM
	BITS 4-0:	INCLINE (0–31°)
BYTE 11	BIT 7	SPEED Decimal Indicator 1 = decimal point 0 = no decimal
	BITS 6-0	SPEED MSB
BYTE 12	BITS 7-0	SPEED LSB (0–32767) or (0.0 to 3276.7)
BYTE 13	BITS 7-0:	CALORIES x 10 (0 TO 2550)
BYTE 14	BIT 7-6:	Distance Type 0=Kilometers 1=Meters 2=Miles 3=feet
	BIT 5	Distance Decimal Indicator 1 = decimal present
	BITS 4-0	DISTANCE MSB
BYTE 15:	BITS 7-0:	DISTANCE LSB (0 to 8191 or 0.0 to 819.1)
BYTE 16	BITS 7-0:	DURATION (0–255mins.)
BYTE 17:	BITS 7-4:	Duration type 0 = seconds 1=minutes 2=hours
	BITS 3-0	#CHARACTERS IN EQUIPMENT NAME (1–6) + 1
BYTE 18	222 11111	
BYTE 19	4 33333 22	
BYTE 20	5555 4444	
BYTE 21	77 66666 5	
BYTE 22	88888 777	
BYTE 23	AAA 99999	
BYTE 24	C BBBBB AA	
BYTE 25	DDDD CCCC	
BYTE 26	FF EEEEE D	
BYTE 27	GGGGG FFF	

Exercise Group Byte (Byte 1)

For an Aerobic Exercise with embedded exercise name, this command byte base is always 0x06. The exercise set taken from the ex_set field of Table A is trimmed to three bits and inserted in the upper nibble of the Exercise Group Byte bits 0–3.

For example: The exercise belongs to set 2: Exercise Group Byte=0x26

Mechanical Setting Format Byte (Byte 2)

The lower 6 bits in this byte indicate if the corresponding mechanical/ergonomic setting is an Alpha character (A–Z) or numeric character (0–32). The program examines the seat1 through seat7 fields of Table A and if it detects an Alpha character, the corresponding bit in Byte 2 is set. If the character string in the field represents a numeric value, the character string is converted into a number (stored back into its Table A field) and the corresponding bit in Byte 2 is cleared.

NOTE: Those seatx fields that are blank (EMPTY) are set to Alpha with a numeric value of 0 (i.e. corresponding bit set in byte 2 and 0x0 in the setting byte)

Number of Settings Byte (Byte 3 Bits 7–5)

Bits 7–5 of Byte 3 represent the number of non-blank seatx fields in the Table A exercise's record. The program will trim the number to three bits and place them into the bit 7 to 5 position of Byte 3. Bit 7 being MSB.

Setting #1 Byte (Byte 3 Bits 4–0)

If the seat1 field is an Alpha character, the ASCII code is trimmed to the most significant 5-bits and stored in bits 4 to 0 of Byte 3. If the seat1 field was numeric, the converted value (ASCII numeric character to number) is also trimmed to 5-bits and stored in bits 4 to 0 of Byte 3. Bit 4 being MSB.

Remaining Settings Packed Bytes (Bytes 4–7)

NOTE: The seatx values are converted to the appropriate representation for Alpha or number and trimmed to 5 bits.

In a similar fashion as the client's Packed Last Name Bytes, the remaining mechanical/ergonomic setting values are packed into the next 4 bytes. In the following table, the number 2 represents the bits for the converted and trimmed seat2 field value, the number 3 for seat3, number 4 for seat4, etc. The last two bits in Byte 7 are unused and cleared. Any seatx field that was blank (EMPTY) has its bits all cleared (set to 0).

BYTE 4	333 22222	complete #2, partial #3
BYTE 5	5 44444 33	remaining #3, complete #4, partial #5
BYTE 6	6666 5555	remaining #5, partial #6
BYTE 7	XX 77777 6	remaining #6, complete #7
	BIT 6:	Spare set to 0
	BIT 7:	Spare set to 0

Intensity Setting Byte (Byte 8)

The intensity field value of Table A is trimmed to a single byte (valid range for field is 0–255) and stored in Byte 8

Level Setting Format Byte (Byte 9 Bit 7)

Bit 7 of Byte 9 indicates if the level field of Table A holds an Alpha character (A–Z) or a numeric character string (0–127). If the value is alpha, trim the ASCII character byte to 7 bits and set the Bit 7 of Byte 9. If the value is a numeric character string, convert the character string to an actual number, trim to 7 bits and clear Bit 7 or Byte 9.

Level Setting Byte (Byte 9 Bits 6–0)

The trimmed level value from above is stored in Bits 6–0 of Byte 9. Bit 6 being MSB.

Byte (Byte 10 Bits 7–5)

The spd_typ field from Table A contains a code number. Convert the code number as follows, trim to 3 bits and store in Bits 7 to 5 of Byte 10:

spd_typ	Bits 7–5	Units
1	000	mph
2	001	kph
3	010	fpm
4	100	mpm
5	101	spm

Incline Setting Byte (Byte 10 Bits 4–0)

The incline field value from Table A is trimmed to 5 bits and stored to Bits 4 to 0 of Byte 10

Speed Decimal Indicator (Byte 11 Bit 7)

If the speed field from Table A is a value with a fraction (0.1 to 0.9), the program will set Bit 7 of Byte 11 and adjust the actual value as described in the following section.

Speed Setting Bytes (Byte 11 Bits 6–0 & Byte 12)

If the speed field from Table A has a fraction (0.1 to 0.9) the program must first multiply the value by 10 to get rid of the tenths value. If the speed field from Table A is an integer, no multiplication takes place. The product is then trimmed to the 15 bits with the most significant 7 bits stored in Bits 6 to 0 of Byte 11. Bit 6 being MSB. The remaining 8-bits are stored in Byte 12. Bit 7 being MSB.

Calories Setting Byte (Byte 13)

The calories field of Table A is divided by ten with the remainder discarded and the quotient trimmed to 8-bits. The resultant byte is stored to Byte 13. Bit 7 being MSB.

NOTE The calories field's VALID clause prevents a number not a multiple of ten from being entered.

Distance Unit Code Setting Byte (Byte 14 Bits 7–6)

The dis_typ field from Table A contains a code number. Convert the code number as follows, trim to 2 bits and store in Bits 7 to 6 of Byte 14:

dis_typ	Bits 7–6	Units
1	00	km.
2	01	m.
3	10	mi.
4	11	ft.

Distance Decimal Indicator Byte (Byte 14 Bit 5)

If the distance field from Table A is a value with a fraction (0.1 to 0.9), the program will set Bit 5 of Byte 14 and adjust the actual value as described in the following section.

Distance Setting Bytes (Byte 14 Bits 4–0 & Byte 15)

If the distance from Table A has a fraction (0.1 to 0.9) the program must first multiply the value by 10 to get rid of the tenths value. If the distance field from Table A is an integer, no multiplacation takes place. The product is then trimmed to the 13 bits with the most significant 5 bits stored in Bits 4 to 0 of Byte 14. Bit 4 being MSB. The remaining 8-bits are stored in Byte 15. Bit 7 being MSB.

Duration Setting Byte (Byte 16)

The duration field of Table A is trimmed to 8-bits. The resultant byte is stored to Byte 16. Bit 7 being MSB.

Duration Unit Code Setting Byte (Byte 17 Bits 7–4)

The dur_typ field from Table A contains a code number. Convert the code number as follows, trim to 4 bits ant store in Bits 7 to 4 of Byte 17:

dur_typ	Bits 7–4	Units
1	0000	sec.
2	0001	min.
3	0010	hrs.

# of Characters In Exercise Name Setting Byte (Byte 17 Bits 3–0)

The exercise field from Table A is trimmed of leading and training blanks, the number of character are counted and the value is trimmed to 4 bits. The value is then decremented such that 16 actual characters is represented by the number 15, 14 by 13, etc. The adjusted value stored to Bits 3 to 0 of Byte 17. Bit 3 being MSB

Exercise Name Packed Bytes (Bytes 18–27)

In a similar fashion as the client's Packed Last Name Bytes, the embedded exercise name ASCII character values are packed into the next 10 bytes. All ASCII characters are trimmed to 5 bits. the upper 3 bits are added by the DTA software. In the following table, the number 1 represents the bits for the trimmed ASCII value of the exercise name's first character. The number 2 for the second character, A for 10th, etc. Unused or leftover bits are set to 0.

BYTE 18	222 11111	1st Character, partial 2nd
BYTE 19	4 33333 22	remaining 2nd, complete 3rd & partial 4th
BYTE 20	5555 4444	remaining 4th, partial 5th
BYTE 21	77 66666 5	remaining 5th, complete 6th, partial 7th
BYTE 22	88888 777	remaining 7th, complete 8th
BYTE 23	AAA 99999	complete 9th, partial 10th
BYTE 24	C BBBBB AA	remaining 10th, complete 11th, partial 12th
BYTE 25	DDDD CCCC	remaining 12th, partial 13th
BYTE 26	FF EEEEE D	remaining 13th, complete 14th, partial 15th
BYTE 27	GGGGG FFF	remaining 15th, complete 16th

Aerobic Exercise with okup Table Code Exercise Name

NOTE: The following table lists the bytes in the formatted exercise block starting with the arbitrary number 1. The actual byte number in the DTA Data Package Buffer will depend upon its actual position amongst the other re-formatted exercise blocks.

BYTE 1	Exercise Group	
BYTE 2	BIT 7:	Spare
	BIT 6:	Setting 7 1 = ALPHA 0 = NUMERIC
	BIT 5:	Setting 6 1 = ALPHA 0 = NUMERIC
	BIT 4:	Setting 5 1 = ALPHA 0 = NUMERIC
	BIT 3:	Setting 4 1 = ALPHA 0 = NUMERIC
	BIT 2:	Setting 3 1 = ALPHA 0 = NUMERIC
	BIT 1:	Setting 2 1 = ALPHA 0 = NUMERIC
	BIT 0:	Setting 1 1 = ALPHA 0 = NUMERIC
BYTE 3	BITS 7–5:	# OF SETTINGS (0–7)
	BITS 4–0:	Setting #1 (0–32 or A–Z)
BYTE 4	333 22222	Setting #2 & #3
BYTE 5	5 44444 33	Setting #3, #4 & #5
BYTE 6	6666 5555	Setting #5 & #6
BYTE 7	XX 77777 6	Setting #6 & #7
	BIT 6:	Spare
	BIT 7:	Spare
BYTE 8	BITS 7–0:	INTENSITY beats per minute (0–255)
BYTE 9	BIT 7:	1 = LEVEL IS ALPHA 0=LEVEL IS NUMERIC
	BITS 6–0:	LEVEL (0–127 or A–Z)
BYTE 10	BITS 7–5:	RATE: 0=MPH 1=KPH 2=FPM 3=MPM 4=SPM
	BITS 4–0:	INCLINE (0–31°)
BYTE 11	BIT 7	SPEED Decimal Indicator 1 = decimal point 0 = no decimal
	BITS 6–0	SPEED MSB
BYTE 12	BITS 7–0	SPEED LSB (0–32767) or (0.0 to 3276.7)
BYTE 13	BITS 7–0:	CALORIES x 10 (0 TO 2550)
BYTE 14	BIT 7–6:	Distance Type 0=Kilometers 1=Meters 2=Miles 3=feet
	BIT 5	Distance Decimal Indicator 1 = decimal present
	BITS 4–0	DISTANCE MSB
BYTE 15:	BITS 7–0:	DISTANCE LSB (0 to 8191 or 0.0 to 819.1)
BYTE 16	BITS 7–0:	DURATION (0–255 mins.)
BYTE 17:	BITS 7–4:	Duration type 0 = seconds 1=minutes 2=hours
	BITS 3–0	Set to 0
BYTE 18:	Lookup Table # LSB	
BYTE 19:	Lookup Table # MSB	

Exercise Group Byte (Byte 1)

For an Aerobic Exercise with Lookup Table Code, this command byte base is always 0x07. The exercise set taken from the ex_set field of Table A is trimmed to three bits and inserted in the upper nibble of the Exercise Group Byte bits **0–3**.

For example: The exercise belongs to set **3**: Exercise Group Byte=0x37

Mechanical Setting Format Byte (Byte 2)

The lower 6 bits in this byte indicate if the corresponding mechanical/ergonomic setting is an Alpha character (A–Z) or numeric character (0–32). The program examines the seat1 through seat7 fields of Table A and if it detects an Alpha character, the corresponding bit in Byte 2 is set. If the character string in the field represents a numeric value, the character string is converted into a number (stored back into its Table A field) and the corresponding bit in Byte 2 is cleared.

NOTE: Those seatx fields that are blank (EMPTY) are set to Alpha with a numeric value of 0 (i.e. corresponding bit set in byte 2 and 0x0 in the setting byte)

Number of Settings Byte (Byte 3 Bits **7–5**)

Bits **7–5** of Byte 3 represent the number of non-blank seatx fields in the Table A exercise's record. The program will trim the number to three bits and place them into the bit **7** to **5** position of Byte 3. Bit **7** being MSB.

Setting #1 Byte (Byte 3 Bits **4–0**)

If the seat1 field is an Alpha character, the ASCII code is trimmed to the most significant 5-bits and stored in bits **4** to **0** of Byte 3. If the seat1 field was numeric, the converted value (ASCII numeric character to number) is also trimmed to 5-bits and stored in bits **4** to **0** of Byte 3. Bit **4** being MSB.

Remaining Settings Packed Bytes (Bytes 4–7)

NOTE: The seatx values are converted to the appropriate representation for Alpha or number and trimmed to 5 bits.

In a similar fashion as the client's Packed Last Name Bytes, the remaining mechanical/ergonomic setting values are packed into the next 4 bytes. In the following table, the number 2 represents the bits for the converted and trimmed seat2 field value, the number 3 for seat3, number 4 for seat4, etc. The last two bits in Byte 7 are unused and cleared. Any seatx field that was blank (EMPTY) has its bits all cleared (set to 0).

BYTE 4	333 22222	complete #2, partial #3
BYTE 5	5 44444 33	remaining #3, complete #4, partial #5
BYTE 6	6666 5555	remaining #5, partial #6
BYTE 7	XX 77777 6	remaining #6, complete #7
	BIT 6:	Spare set to 0
	BIT 7:	Spare set to 0

Intensity Setting Byte (Bytes 8)

The intensity field value of Table A is trimmed to a single byte (valid range for field is 0–255) and stored in Byte 8

Level Setting Format Byte (Byte 9 Bit **7**)

Bit **7** of Byte 9 indicates if the level field of Table A holds an Alpha character (A–Z) or a numeric character string (0–127). If the value is alpha, trim the ASCII character byte to 7 bits and set the Bit **7** of Byte 9. If the value is a numeric character string, convert the character string to an actual number, trim to **7** bits and clear Bit 7 or Byte 9.

Level Setting Byte (Byte 9 Bits **6–0**)

The trimmed level value from above is stored in Bits **6–0** of Byte 9. Bit **6** being MSB.

Byte (Byte 10 Bits **7–5**)

The spd_typ field from Table A contains a code number. Convert the code number as follows, trim to 3 bits and store in Bits **7** to **5** of Byte 10:

spd_typ	Bits 7–5	Units
1	000	mph
2	001	kph
3	010	fpm
4	100	mpm
5	101	spm

Incline Setting Bytes)Byte 10 Bits **4–0**)

The incline field value from Table A is trimmed to 5 bits and stored to Bits **4** to **0** of Byte 10

Speed Decimal Indicator Byte (Byte 11 Bit **7**)

If the speed field from Table A is a value with a fraction (0.1 to 0.9), the program will set Bit **7** of Byte 11 and adjust the actual value as described in the following section.

Speed Setting Bytes (Bytes 11 Bits **6–0** & Byte 12)

If the speed field from Table A has a fraction (0.1 to 0.9) the program must first multiply the value by 10 to get rid of the tenths value. If the speed field from Table A is an integer, no multiplication takes place. The product is then trimmed to the 15 bits with the most significant 7 bits stored in Bits **6** to **0** of Byte 11. Bit **6** being MSB. The remaining 8-bits are stored in Byte 12. Bit **7** being MSB.

Calories Setting Byte (Byte 13)

The calories field of Table A is divided by ten with the remainder discarded and the quotient trimmed to 8-bits. The resultant byte is stored to Byte 13. Bit **7** being MSB.

NOTE The calorie field's VALID clause prevents a number not a multiple of ten from being entered.

Distance Unit Code Setting Byte (Byte 14 Bits 7–6)

The dis_typ field from Table A contains a code number. Convert the code number as follows, trim to 2 bits and store in Bits 7 to 6 of Byte 14:

dis_typ	Bits 7–6	Units
1	00	km.
2	01	m.
3	10	mi.
4	11	ft.

Distance Decimal Indicator Byte (Byte 14 Bit 5)

If the distance field from Table A is a value with a fraction (0.1 to 0.9), the program will set bit 5 of Byte 14 and adjust the actual value as described in the following section.

Distance Setting Bytes (Byte 14 Bits 4–0 & Byte 15)

If the distance field from Table A has a fraction (0.1 to 0.9) the program must first multiply the value by 10 to get rid of the tenths value. If the distance field from Table A is an integer, no multiplication takes place. The product is then trimmed to the 13 bits with the most significant 5 bits stored in Bits 4 to 0 of Byte 14. Bit 4 being MSB. The remaining 8-bits are stored in Byte 15. Bit 7 MSB.

Duration Setting Byte (Byte 16)

The duration field of Table A is trimmed to 8-bits. The resultant byte is stored to Byte 16. Bit 7 being MSB.

Duration Unit Code Setting Byte (Byte 17 Bits 7–4)

The dur_typ field from Table A contains a code number. Convert the code number as follows, trim to 4 bits and store in Bits 7 to 4 of Byte 17:

dur_typ	Bits 7–4	Units
1	0000	sec.
2	0001	min.
3	0010	hrs.

Lookup Table # Bytes (Bytes 18–19)

The table_no field from Table A contains a lookup table code number. Trim to number to two bytes and store the LSB in Byte 18 and the MSB in Byte 19.

Anaerobic Exercise W/embedded Exercise Name

NOTE: The following table lists the bytes in the formatted exercise block starting with the arbitrary number 1. The actual byte number in the DTA Data Package Buffer will depend upon its actual position amongst the other re-formatted exercise blocks.

BYTE 1	Exercise Group	
BYTE 2	BIT 7:	Spare
	BIT 6:	Setting 7 1 = ALPHA 0 = NUMERIC
	BIT 5:	Setting 6 1 = ALPHA 0 = NUMERIC
	BIT 4:	Setting 5 1 = ALPHA 0 = NUMERIC
	BIT 3:	Setting 4 1 = ALPHA 0 = NUMERIC
	BIT 2:	Setting 3 1 = ALPHA 0 = NUMERIC
	BIT 1:	Setting 2 1 = ALPHA 0 = NUMERIC
	BIT 0:	Setting 1 1 = ALPHA 0 = NUMERIC
BYTE 3	BITS 7–5:	# OF SETTINGS (0–7)
	BITS 4–0:	Setting #1 (0–32 or A–Z)
BYTE 4	333 22222	Setting #2 & #3
BYTE 5	5 44444 33	Setting #3, #4 & #5
BYTE 6	6666 5555	Setting #5 & #6
BYTE 7	XX 77777 6	Setting #6 & #7
	BIT 6:	Spare
	BIT 7:	Spare
BYTE 8	BIT 7:	0 = POUNDS 1 = PLATES
	BIT 6:	1 = Decimal in pounds/Plate 0 = No decimal
	BITS 5–0:	POUNDS/PLATES MSB
BYTE 9	BITS 7–0	POUNDS/PLATES (0–16383 (1638.3)) LSB
BYTE 10	BITS 7–4	Duration type 0 = seconds 1 = minutes 2 = hours
	BITS 3–0	Spare
BYTE 11	BITS 7–0	DURATION (0–255)
BYTE 12	BITS 7–0	REPS (0–255)
BYTE 13	BITS 7–4:	SETS (0–15)
	BITS 3–0	# CHARACTERS IN EQUIPMENT NAME (1–16) +1
BYTE 14	222 11111	
BYTE 15	4 33333 22	
BYTE 16	5555 4444	
BYTE 17	77 66666 5	
BYTE 18	88888 777	
BYTE 19	AAA 99999	
BYTE 20	C BBBBB AA	
BYTE 21	DDDD CCCC	
BYTE 22	FF EEEEE D	
BYTE 23	GGGGG FFF	

Exercise Group Byte (Byte 1)

For an Anaerobic Exercise with embedded exercise name, this command byte base is always 0x08. The exercise set taken from the ex_set field of Table A is trimmed to three bits and inserted in the upper nibble of the Exercise Group Byte bits 0–3.

For example: The exercise belongs to set 5: Exercise Group Byte=0x58

Mechanical Setting Format Byte (Byte 2)

The lower 6 bits in this byte indicate if the corresponding mechanical/ergonomic setting is an Alpha character (A–Z) or numeric character (0–32). The program examines the seat1 through seat7 fields of Table A and if it detects an Alpha character, the corresponding bit in Byte 2 is set. If the character string in the field represents a numeric value, the character string is converted into a number (stored back into its Table A field) and the corresponding bit in Byte 2 is cleared.

NOTE: Those seatx fields that are blank (EMPTY) are set to Alpha with a numeric value of 0 (i.e. corresponding bit set in byte 2 and 0x0 in the setting byte)

Number of Settings Byte (Byte 3 Bits 7–5)

Bits 7–5 of Byte 3 represent the number of non-blank seatx fields in the Table A exercise's record. The program will trim the number to three bits and place them into the bit 7 to 5 position of Byte 3. Bit 7 being MSB.

Setting #1 Byte (Byte 3 Bits 4–0)

If the seat1 field is an Alpha character, the ASCII code is trimmed to the most significant 5-bits and stored in bits 4 to 0 of Byte 3. If the seat1 field was numeric, the converted value (ASCII numeric character to number) is also trimmed to 5-bits and stored in bits 4 to 0 of Byte 3. Bit 4 being MSB.

Remaining Settings Packed Bytes (Bytes 4–7)

NOTE: The seatx values are converted to the appropriate representation for Alpha or number and trimmed to 5 bits.

In a similar fashion as the client's Packed Last Name Bytes, the remaining mechanical/ergonomic setting values are packed into the next 4 bytes. In the following table, the number 2 represents the bits for the converted and trimmed seat2 field value, the number 3 for seat3, number 4 for seat4, etc. The last two bits in Byte 7 are unused and cleared. Any seatx field that was blank (EMPTY) has its bits all cleared (set to 0).

BYTE 4	333 22222	complete #2, partial #3
BYTE 5	5 44444 33	remaining #3, complete #4, partial #5

-continued

BYTE 6	6666 5555	remaining #5, partial #6
BYTE 7	XX 77777 6	remaining #6, complete #7
	BIT 6:	Spare set to 0
	BIT 7:	Spare set to 0

Pounds or Plates Type Setting Byte (Byte 8 Bit 7)

The lbs_plt field value of Table A contains a numeric value. If=1 then pounds are selected and Bit 7 of Byte 8 is cleared (0). If the value of lbs_plt is 0, set Bit 7 of Byte 8 to 1

Pounds/Plates Decimal Indicator Byte (Byte 8 Bit 6)

If the pounds field from Table A is a value with a fraction (0.1 to 0.9), the program will set Bit 6 of Byte 8 and adjust the actual value as described in the following section.

Pounds/Plates Setting Bytes (Byte 8 Bits 5–0 & Byte 9)

If the pounds field from Table A has a fraction (0.1 to 0.9) the program must first multiply the value by 10 to get rid of the tenths value. If the pounds field from Table A is an integer, no multiplication takes places The product is then trimmed to the 14 bits with the most significant 6 bits stored in Bits 5 to 0 of Byte 8. Bit 5 being MSB. The remaining 8-bits are stored in Byte 9. Bit 7 being MSB.

Duration Unit Code Setting Byte (Byte 10 Bits 7–4)

The dur_typ field from Table A contains a code number. Convert the code number as follows, trim to 4 bits and store in Bits 7 to 4 of Byte 10:

dur_typ	Bits 7–4	Units
1	0000	sec.
2	0001	min.
3	0010	hrs.

NOTE: Bits 3–0 of Byte 10 are set to 0

Duration Setting Byte (Byte 11)

The duration field of Table A is trimmed to 8-bits. The resultant byte is stored to Byte 11. Bit 7 being MSB.

Repetitions Setting Byte 12)

The reps field of Table A is trimmed to 8-bits. The resultant byte is stored to Byte 12. Bit 7 being MSB.

Sets Setting Byte (Byte 13 Bits 7–4)

The sets field of Table A is trimmed to 4-bits. The resultant value is stored to Bits 7 to 4 of Byte 13. Bit 7 being MSB.

# of characters in Exercise Name Setting Byte (Byte 13 Bits 3–0)

The exercise field from Table A is trimmed of leading and training blanks, the number of character are counted and the value is trimmed to 4 bits. The value is then decremented such that 16 actual characters is represented by the number 15, 14 by 13, etc. The adjusted value stored to Bits 3 to 0 of Byte 13. Bit 3 being MSB

Exercise Name Packed Bytes (Bytes 14–23)

In a similar fashion as the client's Packed Last Name Bytes, the embedded exercise name ASCII character values are packed into the next 10 bytes. All ASCII characters are trimmed to 5 bits. the upper 3 bits are added by the DTA software. In the following table, the number 1 represents the bits for the trimmed ASCII value of the exercise name's first character. The number 2 for the second character, A for 10th, etc. Unused or leftover bits are set to 0.

BYTE 14	222 11111	1st Character, partial 2nd
BYTE 15	4 33333 22	remaining 2nd, complete 3rd & partial 4th
BYTE 16	5555 4444	remaining 4th, partial 5th
BYTE 17	77 66666 5	remaining 5th, complete 6th, partial 7th
BYTE 18	88888 777	remaining 7th, complete 8th
BYTE 19	AAA 99999	complete 9th, partial 10th
BYTE 20	C BBBBB AA	remaining 10th, complete 11th, partial 12th
BYTE 21	DDDD CCCC	remaining 12th, partial 13th
BYTE 22	FF EEEEE D	remaining 13th, complete 14th, partial 15th
BYTE 23	GGGGG FFF	remaining 15th, complete 16th

Anaerobic Exercise with Lookup Table Code Exercise Name

NOTE: The following table slits the bytes in the formatted exercise block starting with the arbitrary number 1. The actual byte number in the DTA Data Package Buffer will depend upon its actual position amongst the other re-formatted exercise blocks.

BYTE 1	Exercise Group	
BYTE 2	BIT 7:	Spare
	BIT 6:	Setting 7 1 = ALPHA 0 = NUMERIC
	BIT 5:	Setting 6 1 = ALPHA 0 = NUMERIC
	BIT 4:	Setting 5 1 = ALPHA 0 = NUMERIC
	BIT 3:	Setting 4 1 = ALPHA 0 = NUMERIC
	BIT 2:	Setting 3 1 = ALPHA 0 = NUMERIC
	BIT 1:	Setting 2 1 = ALPHA 0 = NUMERIC
	BIT 0:	Setting 1 1 = ALPHA 0 = NUMERIC
BYTE 3	BITS 7–5:	# OF SETTINGS (0–7)
	BITS 4–0:	Setting #1 (0–32 or A–Z)
BYTE 4	333 22222	Setting #2 & #3
BYTE 5	5 44444 33	Setting #3, #4 & #5
BYTE 6	6666 5555	Setting #5 & #6
BYTE 7	XX 77777 6	Setting #6 & #7
	BIT 6:	Spare
	BIT 7:	Spare
BYTE 8	BIT 7:	0 = POUNDS 1 = PLATES
	BIT 6:	1 = Decimal in pounds/Plate 0 = No decimal
	BITS 5–0:	POUNDS/PLATES MSB
BYTE 9	BITS 7–0	POUNDS/PLATES (0–16383 (1638.3)) LSB
BYTE 10	BITS 7–0	REPS (0–255)
BYTE 11	BITS 7–4:	SETS (0–15)
	BITS 3–0	Duration type 0 = seconds 1 = minutes 2 = hours
BYTE 12	BITS 7–0	DURATION (0–255)
BYTE 13	LOOKUP TABLE # LSB	
BYTE 14	LOOKUP TABLE # MSB	

Exercise Group Byte (Byte 1)

For an Anaerobic Exercise with Lookup Table Code, this command byte base is always 0x09. The exercise set taken from the ex_set field of Table A is trimmed to three bits and inserted in the upper nibble of the Exercise Group Byte bits 0–3.

For example: The exercise belongs to set 3: Exercise Group Byte=0x39

Mechanical Setting Format Byte (Byte 2)

The lower 6 bits in this byte indicate if the corresponding mechanical/ergonomic setting is an Alpha character (A–Z) or numeric character (0–32). The program examines the seat1 through seat7 fields of Table A and if it detects an Alpha character, the corresponding bit in Byte 2 is set. If the character string in the field represents a numeric value, the character string is converted into a number (stored back into its Table A field) and the corresponding bit in Byte 2 is cleared.

NOTE: Those seatx fields that are blank (EMPTY) are set to Alpha with a numeric value of 0 (i.e. corresponding bit set in byte 2 and 0x0 in the setting byte)

Number of Settings Byte (Byte 3 Bits 7–5)

Bits 7–5 of Byte 3 represent the number of non-blank seatx fields in the Table A exercise's record. The program

will trim the number to three bits and place them into the bit 7 to 5 position of Byte 3. Bit 7 being MSB.

Setting #1 Byte (Byte 3 Bits 4–0)

If the seat1 field is an Alpha character, the ASCII code is trimmed to the most significant 5-bits and stored in bits 4 to 0 of Byte 3. If the seat1 field was numeric, the converted value (ASCII numeric character to number) is also trimmed to 5-bits and stored in bits 4 to 0 of Byte 3. Bit 4 being MSB.

Remaining Settings Packed Bytes (Bytes 4–7)

NOTE: The seatx values are converted to the appropriate representation for Alpha or number and trimmed to 5 bits.

In a similar fashion as the client's Packed Last Name Bytes, the remaining mechanical/ergonomic setting values are packed into the next 4 bytes. In the following table, the number 2 represents the bits for the converted and trimmed seat2 field value, the number 3 for seat3, number 4 for seat4, etc. The last two bits in Byte 7 are unused and cleared. Any seatx field that was blank (EMPTY) has its bits all cleared (set to 0).

BYTE 4	333 22222	complete #2, partial #3
BYTE 5	5 44444 33	remaining #3, complete #4, partial #5
BYTE 6	6666 5555	remaining #5, partial #6
BYTE 7	XX 77777 6	remaining #6, complete #7
	BIT 6:	Spare set to 0
	BIT 7:	Spare set to 0

Pounds or Plates Type Setting Byte (Byte 8 Bit 7)

The lbs_plt field value of Table A contains a numeric value. If=1 then pounds are selected and Bit 7 of Byte 8 is cleared (0). If the value of lbs_plt is 0, set Bit 7 of Byte 8 to 1

Pounds/Plates Decimal Indicator Byte (Byte 8 Bit 6)

If the pounds field from Table A is a value with a fraction (0.1 to 0.9), the program will set Bit 6 of Byte 8 and adjust the actual value as described in the following section.

Pounds/Plates Setting Bytes (Byte 8 Bits 5–0 & Byte 9)

If the pounds field from Table A has a fraction (0.1 to 0.9) the program must first multiply the value by 10 to get rid of the tenths value. If the pounds field from Table A is an integer, no multiplication takes places The product is then trimmed to the 14 bits with the most significant 6 bits stored in Bits 5 to 0 of Byte 8. Bit 5 being MSB. The remaining 8-bits are stored in Byte 9. Bit 7 being MSB.

Repetitions Setting Byte (Byte 10)

The reps field of Table A is trimmed to 8-bits. The resultant byte is stored to Byte 10. Bit 7 being MSB.

Sets setting Byte (Byte 11 Bits 7–4)

The sets field of Table A is trimmed to 4-bits. The resultant value is stored to Bits 7 to 4 of Byte 11. Bit 7 being MSB.

Duration Unit Code Setting Byte (Byte 11 Bits 3–0)

The dur_typ field from Table A contains a code number. Convert the code number as follows, trim to 4 bits and store in Bits 3 to 0 of Byte 11:

dur_typ	Bits 3–0	Units
1	0000	sec.
2	0001	min.
3	0010	hrs.

Duration Setting Byte (Byte 12)

The duration field of Table A is trimmed to 8-bits. The resultant byte is stored to Byte 12. Bit 7 being MSB.

Lookup Table # Bytes (Bytes 13–14)

The table_no field from Table A contains a lookup table code number. Trim to number to two bytes and store the LSB in Byte 13 and the MSB in Byte 14.

Stretch Exercise w/Embedded Exercise Name

NOTE: The following table slits the bytes in the formatted exercise block starting with the arbitrary number 1. The actual byte number in the DTA Data Package Buffer will depend upon its actual position amongst the other re-formatted exercise blocks.

BYTE 1	Exercise Group	
BYTE 2	BITS 7–0:	DURATION (0–255)
BYTE 3	BITS 7–4:	Duration type 0 = seconds 1=minutes 2=hours
	BITS 3–0:	Spare
BYTE 4	BIT 7–0:	REPS (0–255)
BYTE 5	BITS 7–4:	SETS (0–15)
	BITS 3–0	# CHARACTERS IN EQUIPMENT NAME (1–16) +1
BYTE 6	222 11111	
BYTE 7	4 33333 22	
BYTE 8	5555 4444	
BYTE 9	77 66666 5	
BYTE 10	88888 777	
BYTE 11	AAA 99999	
BYTE 12	C BBBBB AA	
BYTE 13	DDDD CCCC	
BYTE 14	FF EEEEE D	
BYTE 15	GGGGG FFF	

Exercise Group Byte (Byte 1)

For an Stretch Exercise with embedded exercise name, this command byte base is always 0x0A. The exercise set taken from the ex_set field of Table A is trimmed to three bits and inserted in the upper nibble of the Exercise Group Byte bits 0–3.

For example: The exercise belongs to set 3: Exercise Group Byte=0x3A

Duration Setting Byte (Byte 2)

The duration field of Table A is trimmed to 8-bits. The resultant byte is stored to Byte 2. Bit 7 being MSB.

Duration Unit Code Setting Byte (Byte 3 Bits 7–4)

The dur_typ field from Table A contains a code number. Convert the code number as follows, trim to 4 bits and store in Bits 7 to 4 of Byte 3:

dur_typ	Bits 7–4	Units
1	0000	sec.
2	0001	min.
3	0010	hrs.

NOTE: Bits 3–0 of Byte 10 are set to 0

Repetitions Setting Byte (Byte 4)

The reps field of Table A is trimmed to 8-bits. The resultant byte is stored to Byte 4. Bit 7 being MSB.

Sets setting Byte (Byte 5 Bits 7–4)

The sets field of Table A is trimmed to 4-bits. The resultant value is stored to Bits 7 to 4 of Byte 5. Bit 7 being MSB.

# of Characters In Exercise Name Setting Byte (Byte 5 Bits 3–0)

The exercise field from Table A is trimmed of leading and training blanks, the number of character are counted and the value is trimmed to 4 bits. The value is then decremented such that 16 actual characters is represented by the number 15, 14 by 13, etc. The adjusted value stored to Bits 3 to 0 of Byte 5. Bit 3 being MSB

Exercise Name Packed Bytes (Bytes 6–15)

In a similar fashion as the client's Packed Last Name Bytes, the embedded exercise name ASCII character values are packed into the next 10 bytes. All ASCII characters are trimmed to 5 bits. the upper 3 bits are added by the DTA software. In the following table, the number 1 represents the

bits for the trimmed ASCII value of the exercise name's first character. The number 2 for the second character, A for 10th, etc. Unused or leftover bits are set to 0.

BYTE 6	222 11111	1st Character, partial 2nd
BYTE 7	4 33333 22	remaining 2nd, complete 3rd & partial 4th
BYTE 8	5555 4444	remaining 4th, partial 5th
BYTE 9	77 66666 5	remaining 5th, complete 6th, partial 7th
BYTE 10	88888 777	remaining 7th, complete 8th
BYTE 11	AAA 99999	complete 9th, partial 10th
BYTE 12	C BBBBB AA	remaining 10th, complete 11th, partial 12th
BYTE 13	DDDD CCCC	remaining 12th, partial 13th
BYTE 14	FF EEEEE D	remaining 13th, complete 14th, partial 15th
BYTE 15	GGGGG FFF	remaining 15th, complete 16th

Stretch Exercise with Lookup Table Code Exercise Name

NOTE: The following table slits the bytes in the formatted exercise block starting with the arbitrary number 1. The actual byte number in the DTA Data Package Buffer will depend upon its actual position amongst the other re-formatted exercise blocks.

BYTE 1	Exercise Group
BYTE 2	BITS 7–0: DURATION (0–255)
BYTE 3	BITS 7–4: Duration type 0 = seconds 1=minutes 2=hours
	BITS 3–0: Spare
BYTE 4	BIT 7–0: REPS (0–255)
BYTE 5	BITS 7–4: SETS (0–15)
	BITS 3–0 spare
BYTE 6	Lookup Table # LSB
BYTE 7	Lookup Table # MSB

Exercise Group Byte (Byte 1)

For an Stretch Exercise with Lookup Table Code, this command byte base is always 0x0B. The exercise set taken from the ex_set field of Table A is trimmed to three bits and inserted int the upper nibble of the Exercise Group Byte bits 0–3.

For example: The exercise belongs to set 7: Exercise Group Byte=0x7B

Duration Setting Byte (Byte 2)

The duration field of Table A is trimmed to 8-bits. The resultant byte is stored to Byte 2. Bit 7 being MSB. Duration Unit Code Setting Byte (Byte (Byte 3 Bits 7–4)

The dur_typ field from Table A contains a code number. Convert the code number as follows, trim to 4 bits and store in Bits 7 to 4 of Byte 3:

dur_typ	Bits 7–4	Units
1	0000	sec.
2	0001	min.
3	0010	hrs.

NOTE: Bits 3–0 of Byte 10 are set to 0

Repetitions Setting Byte (Byte 4)

The reps field of Table A is trimmed to 8-bits. The resultant byte is stored to Byte 4. Bit 7 being MSB.

Sets Setting Byte (Byte 5 Bits 7–4)

The sets field of Table A is trimmed to 4-bits. The resultant value is stored to Bits 7 to 4 of Byte 5. Bit 7 being MSB.

NOTE: Bits 3–0 of Byte are set to 0

Lookup Table # Bytes (Bytes 6–7)

The table_no field from Table A contains a lookup table code number. Trim to number to two bytes and store the LSB in Byte 6 and the MSB in Byte 7.

Footer Byte

After all the exercise records in Table A have been compiled into exercise blocks and added to the DTA Data Package Buffer, the Footer Byte is appended to tell the DTA that there are no more exercises to process.

Footer Byte=0x04

Checksum

Following the addition of the Footer Byte, the program will count the number of bytes starting with the Header Block's Header Command and including the Footer Byte. IF THE NUMBER OF BYTES IS GREATER THAN 960 then display the bytes.scx alert screen:

Pressing the OK button returns the user back to the screen from which the Program DTA procedure was called from.

The user must then edit the workout routine either reducing the number of exercises or replacing some of the user defined exercise names with table lookup exercises to reduce the size of exercise blocks.

Assuming the byte count was not excessive, the program starts with Header Block's Header Command performs a Modulo-256 checksum up to and including the Footer Byte. The resultant byte (all carries out of the byte are ignored) is appended to the end of the DTA Data Package Buffer (after the footer byte). The Package is now ready to be transmitted to the DTA unit.

Transmitting the Package

By operational definition, a DTA unit has been previously loaded into its Programming Stand and is initialized awaiting a data transmission. The DTA displays:

"READY TO PROGRAM"

The Program DTA Procedure will then transmit the DTA Data Package Buffer, starting with the Header Block's Header Command byte and ending with the checksum byte through the computer's COM1 port at 9600 BAUD with 8 data bits, 1 start bit, 1 stop bit and no parity.

Upon completion of the transmission, the program will sound the computer's bell and display the following:

DTA PROGRAMMED

The Program DTA Procedure then exits back to the screen from which it was called.

NOTE: Any transmission difficulties are handled either by Windows or if at the DTA end, the DTA's communication software. In either case the user should be offered the option or retrying or aborting.

Receive From DTA Subfunction

The only way to activate the Receive From DTA Subfunction is using the DTA→Receive DTA menu option from the Workout Builder Screen or the Download DTA push button from the Automatic Mode screen.

The basic procedure is a reversal of the Transmit to DTA Subfunction. The data package received from the DTA is in the exact format as the package transmitted to it. The only difference being some changes in the embedded parameter values and some of the command header bytes change to reflect the completion status of the exercise

DTA Download Command

By operational definition, a DTA unit has been returned to the Programming Stand and has been placed in a state such that transmission of its data is possible. The Receive From DTA Subfunction must first transmit the download command to the DTA. This is a single byte as follows:

DTA Download Request=0x03

Immediately following this transmission, the computer must be readied to receive 961 bytes from the DTA into a buffer where the Program DTA Procedure code can have access to it.

Checksum Test

The Receive From DTA Subfunction will first checksum the received data package, starting at the beginning of the

buffer until the Footer Byte is detected. The checksum is compared against the checksum embedded as the byte after the Footer in the received data package. Following successful checksumming the buffer is parsed and the data elements extracted and stored in the appropriate client's CWH table.

In order for the checksumming code to detect the Footer byte, it must count bytes and look for the Footer byte code 0x04 in an expected byte position in the received DTA data buffer. A counting algorithm is used to determine what byte number in the buffer the Footer resides in. Once known a checksum can be performed up to and including the Footer byte.

The counting algorithm executes as follows:

1. Start with a byte count=19
2. Skip the first 19 Header Block Bytes
3. Mask off the upper nibble of the next byte. If it then=0x6 look at the 16th byte after this one then:

if the lower 4 bits of the byte = 0xO then add 19 to current byte count
if the lower 4 bits of the byte = 0x1 then add 20 to current byte count
if the lower 4 bits of the byte = 0x2 then add 20 to current byte count
if the lower 4 bits of the byte = 0x3 then add 21 to current byte count
if the lower 4 bits of the byte = 0x4 then add 22 to current byte count
if the lower 4 bits of the byte = 0x5 then add 22 to current byte count
if the lower 4 bits of the byte = 0x6 then add 23 to current byte count
if the lower 4 bits of the byte = 0x7 then add 23 to current byte count
if the lower 4 bits of the byte = 0x8 then add 24 to current byte count
if the lower 4 bits of the byte = 0x9 then add 25 to current byte count
if the lower 4 bits of the byte = 0xA then add 25 to current byte count
if the lower 4 bits of the byte = 0xB then add 26 to current byte count
if the lower 4 bits of the byte = 0xC then add 27 to current byte count
if the lower 4 bits of the byte = 0xD then add 27 to current byte count
if the lower 4 bits of the byte = 0xE then add 28 to current byte count
if the lower 4 bits of the byte = 0xF then add 28 to current byte count

OR

Mask off the upper nibble of the next byte. If it = 0x8 look at the 12th byte after this one then:
if the lower 4 bits of the byte = 0x0 then add 15 to current byte count
if the lower 4 bits of the byte = 0x1 then add 16 to current byte count
if the lower 4 bits of the byte = 0x2 then add 16 to current byte count
if the lower 4 bits of the byte = 0x3 then add 17 to current byte count
if the lower 4 bits of the byte = 0x4 then add 18 to current byte count
if the lower 4 bits of the byte = 0x5 then add 18 to current byte count
if the lower 4 bits of the byte = 0x6 then add 19 to current byte count
if the lower 4 bits of the byte = 0x7 then add 19 to current byte count
if the lower 4 bits of the byte = 0x8 then add 20 to current byte count
if the lower 4 bits of the byte = 0x9 then add 21 to current byte count
if the lower 4 bits of the byte = 0xA then add 21 to current byte count
if the lower 4 bits of the byte = 0xB then add 22 to current byte count
if the lower 4 bits of the byte = 0xC then add 23 to current byte count
if the lower 4 bits of the byte = 0xD then add 23 to current byte count
if the lower 4 bits of the byte = 0xE then add 24 to current byte count
if the lower 4 bits of the byte = 0xF then add 24 to current byte count

OR

Mask off the upper nibble of the next byte. If it = 0xA look at the 4th byte after this one then:
if the lower 4 bits of the byte = 0x0 then add 7 to current byte count
if the lower 4 bits of the byte = 0x1 then add 8 to current byte count
if the lower 4 bits of the byte = 0x2 then add 8 to current byte count
if the lower 4 bits of the byte = 0x3 then add 9 to current byte count
if the lower 4 bits of the byte = 0x4 then add 10 to current byte count
if the lower 4 bits of the byte = 0x5 then add 10 to current byte count
if the lower 4 bits of the byte = 0x6 then add 11 to current byte count
if the lower 4 bits of the byte = 0x7 then add 11 to current byte count
if the lower 4 bits of the byte = 0x8 then add 12 to current byte count
if the lower 4 bits of the byte = 0x9 then add 13 to current byte count
if the lower 4 bits of the byte = 0xA then add 13 to current byte count
if the lower 4 bits of the byte = 0xB then add 14 to current byte count
if the lower 4 bits of the byte = 0xC then add 15 to current byte count
if the lower 4 bits of the byte = 0xD then add 15 to current byte count
if the lower 4 bits of the byte = 0xE then add 16 to current byte count
if the lower 4 bits of the byte = 0xF then add 16 to current byte count

OR

Mask off the upper nibble of the next byte. If it = 0x7 add 19 to current byte count

-continued

OR

Mask off the upper nibble of the next byte. If it = 0x9 add 14 to current byte count

OR

Mask off the upper nibble of the next byte. If it = 0xB add 7 to current byte count

OR

Mask off the upper nibble of the next byte. If it = 0x4 then Footer Found!, note the byte number and proceed to Step 5

4. The process is repeated at step #3 until the footer is found or we exceed 960 bytes in which case we have a problem and the following is displayed:
   Error in Determining Checksum
   The processing is aborted and the program drops back to the screen from which it was called. The user may attempt to download again.
5. We now have the starting byte number and the Footer Byte number, perform a Modulo-256 checksum up to and including the Footer Byte, compare this value against the byte found immediately after the Footer byte in the received DTA data buffer. If a mismatch between values is detected, the following is displayed:
   Checksum Error, Try Again
   The processing is aborted and the program drops back to the screen from which it was called. The user may attempt to download again.
   If the checksum was a success, the Receive From DTA Subfunction continues with the Header Block Parsing

Header Block Parsing

The Receive From DTA Subfunction will first parse the received Header Block:

Byte 1	HEADER COMMAND
Byte 2	MEMBER # LSB
Byte 3	MEMBER#
Byte 4	MEMBER#
Byte 5	MEMBER# MSB
Byte 6	# OF CHARACTERS IN NAME
Byte 7	222 11111
Byte 8	4 3333 22
Byte 9	55555 4444
Byte 10	77 66666 5
Byte 11	88888 777
Byte 12	AAA 99999
Byte 13	C BBBBB AA
Byte 14	DDDD CCCC
Byte 15	FF EEEEE D
Byte 16	GGGGG FFF
Byte 17	WEIGHT LSB
Byte 18	WEIGHT MSB
Byte 19	PULSE

The membership number in Bytes 2 to 5 is extracted and stored in a temporary variable memberno. The weight and pulse bytes are also extracted and stored in temporary variables rweight and rpulse. All other bytes in the received Header Block are ignored.

The program the searches all the client database files in the appropriate subdirectory for a record with a member_no field value matching the membership number in memberno extracted from the Header Block. When found, the client's client Workout History filename is computed and the corresponding file in the appropriate subdirectory is opened.

Exercise Block Parsing and Decoding

Following the parsing the Header Block, the Receive From DTA Subfunction will start to decode the Exercise Blocks and create complete CWH records to be appended to the file. The 20th byte in the buffer is the first exercise header command or possibly the Footer byte.

This parsing and decoding process continues until the Footer Byte is detected. When this occurs, the CWH is closed, the following is displayed and the program returns to the screen from which it was called.

DTA Data Received and Recorded

NOTE: Each new CWH record is created with the structure described earlier. The fields are empty when created with the exception of the ex_date field using the current date, the pulse field using the value from rpulse, and the weight field using the value in rweight, for each new record created and appended to the CWH file.

Weight Data Extraction and Recording

The numeric weight value embedded in the received DTA data buffer bytes 17 and 18 needs to be slightly processed before storing in the weight field of the new CWH record. Bit 7 of Byte 18 is examined. If it is set this indicates a decimal (tenths of a pound) exists the remaining numeric value represented by in Bits 6–0 of Byte 18 (MSB) and all of Byte 17 (LSB) is divided by ten and then stored in the weight field of the CWH record. If Bit 7 of Byte 18 is 0, then the number is stored as is.

Aerobic Exercise w/Embedded Exercise Name

If the lower nibble of the exercise block command byte= 0x6, the following bytes belong to an Aerobic Exercise w/Embedded Exercise Name. If the 7th byte's most significant bit (bit—7) is cleared (0) then the completed field of the new CWH record is loaded with a logical true. For all cases of the Aerobic Exercise w/Embedded Name command byte, the following extraction and recording procedure is performed.

NOTE: The following procedure description refers to specific byte numbers. For clarification, the bytes in the described exercise block start with the arbitrary number 1. The actual byte number in the Received DTA Data Package Buffer will depend upon its actual position amongst the other exercise blocks.

BYTE 1	Exercise Group	
BYTE 2	BIT 7:	Spare
	BIT 6:	Setting 7 1 = ALPHA 0 = NUMERIC
	BIT 5:	Setting 6 1 = ALPHA 0 = NUMERIC
	BIT 4:	Setting 5 1 = ALPHA 0 = NUMERIC
	BIT 3:	Setting 4 1 = ALPHA 0 = NUMERIC
	BIT 2:	Setting 3 1 = ALPHA 0 = NUMERIC
	BIT 1:	Setting 2 1 = ALPHA 0 = NUMERIC
	BIT O:	Setting 1 1 = ALPHA 0 = NUMERIC
BYTE 3	BITS 7-5:	# OF SETTINGS (0–7)
	BITS 4-O:	Setting #1 (0–32 or A–Z)
BYTE 4	333 22222	Setting #2 & #3
BYTE 5	5 44444 33	Setting #3, #4 & #5
BYTE 6	6666 5555	Setting #5 & #6
BYTE 7	XX 77777 6	Setting #6 & #7
	BIT 6:	Spare
	BIT 7:	Completed Status 0=completed as prescribed 1 = modified
BYTE 8	BITS 7-O:	INTENSITY beats per minute (0–255)
BYTE 9	BIT 7:	1= LEVEL IS ALPHA 0=LEVEL IS NUMERIC
	BITS 6-O:	LEVEL (O–127 or A–Z)
BYTE 10	BITS 7-5:	RATE: O=MPH 1=KPH 2=FPM 3=MPM 4=SPM
	BITS 4-O:	INCLINE (0–31°)
BYTE 11	BIT 7	SPEED Decimal Indicator 1= decimal point 0 = no decimal
	BITS 6-O	SPEED MSB
BYTE 12	BITS 7-O	SPEED LSB (O–32767) or (O.O to 3276.7)
BYTE 13	BITS 7-O:	CALORIES x 10 (0 TO 2550)
BYTE 14	BIT 7-6:	Distance Type O=Kilometers 1=Meters 2=Miles 3=feet
	BIT 5	Distance Decimal Indicator 1 = decimal present
	BITS 4-O	DISTANCE MSB
BYTE 15:	BITS 7-O:	DISTANCE LSB (0 to 8191 or 0.0 to 819.1)
BYTE 16	BITS 7-O:	DURATION (0–255 mins.)
BYTE 17:	BITS 7-4:	Duration type 0 = seconds 1=minutes 2=hours
	BITS 3-O	# CHARACTERS IN EQUIPMENT NAME (1–16)+1
BYTE 18	222 11111	
BYTE 19	4 33333 22	
BYTE 20	5555 4444	
BYTE 21	77 66666 5	
BYTE 22	88888 777	
BYTE 23	AAA 99999	
BYTE 24	C BBBBB AA	
BYTE 25	DDDD CCCC	
BYTE 26	FF EEEEE D	
BYTE 27	GGGGG FFF	

Exercise Group Byte (Byte 1)

The Exercise Group byte is decoded to determine what exercise group the block represents. The lower nibble is used to determine the exercise group. The resultant decoded value is stored to the ex_group field of the new CWH record as follows:

Command Byte Lower Nibble	ex_group
0x6	Aerobic
0x7	Aerobic
0x8	Anaerobic
0x9	Anaerobic
OxA	Stretch
OxB	Stretch

Mechanical/Ergonomic Settings Related Bytes (Bytes 2–7)

If the HEADER COMMAND=0x02 (Learn Mode) then the Level settings are decoded from Bytes 3 to 7. Extract the bits as depicted earlier. If the setting is designated as alpha by looking at the corresponding bit in Byte 2, then add 0x40 to the extracted bits. The client's CWR file is examined and the exercise record corresponding to the current exercise name (or lookup number) is accessed. The seatx fields are then updated accordingly.

If the HEADER COMMAND=0x00 (Normal Mode) then Bytes 2–7 are ignored.

Intensity Setting Byte (Byte 8)

The numeric value of the intensity byte is stored in the intensity field of the new CWH record.

Level Setting Format Byte (Byte 9 Bit 7)

The level setting format is noted by the program as Bit 7 or Byte 9 set=Alpha, bit cleared=Numeric

Level Setting Byte (Byte 9 Bits 6–0)

The numeric data is extracted from the byte. If the Level Setting Format designates it as a Alpha character, then the data is treated as an ASCII byte (the 8th bit is 0 anyway) and stored to the level field of the new CWH record. If the format specifies the data value as numeric, the data must be converted to characters then stored in the level field of the new CWH record.

Speed Units Code Setting Byte (Byte 10 Bits 7–5)

The Speed Units are not stored in the CWH file and thus ignored

Incline Setting Byte (Byte 10 Bits 4–0)

The incline data is extracted from Bits 4 to 0 from Byte 10 and stored directly to the incline field of the new CWH record.

Speed Decimal Indicator Byte (Byte 11 Bit 7)

The presence of a decimal point in the speed value is noted by examining Bit 7 of Byte 11. A 1=decimal point present, 0=no decimal point.

Speed Setting Bytes (Byte 11 Bits 6–0 & Byte 12)

The speed data is extracted from Byte 11 Bits 6 to 0 (MSB) and Byte 12 (LSB). If the Speed Decimal Indicator indicates a decimal point, the extracted value is divided by 10 and then stored in the speed field of the new CWH record. If there is no decimal point indicated, the extracted value is stored without dividing first.

Calories Setting Byte (Byte 13)

The calories value is extracted from Byte 13 and first multiplied by 10 before storing directly to the calories field of the new CWH record.

Distance Unit Code Setting Byte (Byte 14 Bits 7–6)

The Distance Units are not stored in the CWH file and thus ignored

Distance Decimal Indicator Byte (Byte 14 Bit 5)

The presence of a decimal point in the distance value is noted by examining Bit 5 of Byte 14. A 1=decimal point present, 0=no decimal point.

Distance Setting Bytes (Byte 14 Bits 4–0 & Byte 15)

The distance data is extracted from Byte 14 Bits 4 to 0 (MSB) and Byte 15 (LSB). If the Distance Decimal Indicator indicates a decimal point, the extracted value is divided by 10 and then stored in the distance field of the new CWH record. If there is no decimal point indicated, the extracted value is stored without dividing first.

Duration Setting Byte (Byte 16)

The duration setting is extracted from Byte 16 and stored directly to the duration field of the new CWH record.

Duration Unit Code Setting Byte (Byte 17 Bits 7–4)

The Duration Units are not stored in the CWH file and thus ignored # of Characters In Exercise Name Setting Byte (Byte 17 Bits 3–0)

The number of Characters value is used in the following section to extract the exercise name from the packed bytes.

Exercise Name Packed Bytes (Bytes 18–27)

In a similar fashion as the packing of the exercise name, the embedded exercise name characters are extracted and unpacked.

The ASCII characters have been trimmed to 5 bits before packing and thus when extracted, the upper 3 bits are restored by adding 0x40 to each extracted character.

NOTE: The ASCII space character, 0x20, is packed as 0. When an extracted 5-bit character=0, add 0x20 instead of 0x40.

The program concatenates the exercise name string one character at a time. When the number of characters equals the Number of Characters value extracted earlier, the string is stored to the exercise field of the new CWH record.

BYTE 18	222 11111	1st Character, partial 2nd
BYTE 19	4 33333 22	remaining 2nd, complete 3rd & partial 4th
BYTE 20	5555 4444	remaining 4th, partial 5th
BYTE 21	77 66666 5	remaining 5th, complete 6th, partial 7th
BYTE 22	88888 777	remaining 7th, complete 8th
BYTE 23	AAA 99999	complete 9th, partial 10th
BYTE 24	C BBBBB AA	remaining 10th, complete 11th, partial 12th
BYTE 25	DDDD CCCC	remaining 12th, partial 13th
BYTE 26	FF EEEEE D	remaining 13th, complete 14th, partial 15th
BYTE 27	GGGGG FFF	remaining 15th, complete 16th

ex_set Field Setting

The ex_set of the new CWH record is set by extracting the value from the lower three bits of the upper nibble of the Exercise Group Byte.

Cleanup

The new CWH exercise record is now complete and appended to the CWH file. A new, blank CWH record is made with the ex_date, pulse and weight fields filled first.

Aerobic Exercise with Lookup Table Code

If the lower nibble of the exercise block command byte= 0x7 then the following bytes belong to an Aerobic Exercise with Lookup Table Code Name.

If the 7th byte's most significant bit (bit—7) is cleared (0) then the completed field of the new CWH record is loaded with a logical true.

For all cases of the Aerobic Exercise with Lookup Table Code Name command byte, the following extraction and recording procedure is performed.

NOTE: The following procedure description refers to specific byte numbers. For clarification, the bytes in the described exercise block start with the arbitrary number 1. The actual byte number in the Received DTA Data Package Buffer will depend upon its actual position amongst the other exercise blocks.

BYTE 1	Exercise Group	
BYTE 2	BIT 7:	Spare
	BIT 6:	Setting 7 1 = ALPHA 0 = NUMERIC
	BIT 5:	Setting 6 1 = ALPHA 0 = NUMERIC
	BIT 4:	Setting 5 1 = ALPHA 0 = NUMERIC
	BIT 3:	Setting 4 1 = ALPHA 0 = NUMERIC
	BIT 2:	Setting 3 1 = ALPHA 0 = NUMERIC
	BIT 1:	Setting 2 1 = ALPHA 0 = NUMERIC
	BIT 0:	Setting 1 1 = ALPHA 0 = NUMERIC
BYTE 3	BITS 7–5:	# OF SETTINGS (0–7)
	BITS 4–0:	Setting #1 (0–32 or A–Z)
BYTE 4	333 22222	Setting #2 & #3
BYTE 5	5 44444 33	Setting #3, #4 & #5
BYTE 6	6666 5555	Setting #5 & #6
BYTE 7	XX 77777 6	Setting #6 & #7
	BIT 6:	Spare

-continued

	BIT 7:	Completed Status 0 = completed as prescribed 1 = modified
BYTE 8	BITS 7–0:	INTENSITY beats per minute (0–255)
BYTE 9	BIT 7:	1 = LEVEL IS ALPHA 0 = LEVEL IS NUMERIC
	BITS 6–0:	LEVEL (0–127 or A–Z)
BYTE 10	BITS 7–5:	RATE: 0 = MPH 1 = KPH 2 = FPM 3 = MPM 4 = SPM
	BITS 4–0:	INCLINE (0–31°)
BYTE 11	BIT 7	SPEED Decimal indicator 1= decimal point 0 = no decimal
	BITS 6–0	SPEED MSB
BYTE 12	BITS 7–0	SPEED LSB (0–32767) or (0.0 to 3276.7)
BYTE 13	BITS 7–0:	CALORIES x 10 (0 TO 2550)
BYTE 14	BIT 7–6:	Distance Type 0 = Kilometers 1 = Meters 2 = Miles 3 = feet
	BIT 5	Distance Decimal Indicator 1 = decimal present
	BITS 4–0	DISTANCE MSB
BYTE 15:	BITS 7–0:	DISTANCE LSB (0 to 8191 or 0.0 to 819.1)
BYTE 16	BITS 7–0:	DURATION (0–255 mins.)
BYTE 17:	BITS 7–4:	Duration type 0 = seconds 1 = minutes 2 = hours
	BITS 3–0	spare
BYTE 18	Lookup Table # LSB	
BYTE 19	Lookup Table # MSB	

Exercise Group Byte (Byte 1)

The Exercise Group byte is decoded to determine what exercise group the block represents. The lower nibble is used to determine the exercise group. The resultant decoded value is stored to the ex_group field of the new CWH record as follows:

Command Byte Lower Nibble	ex_group
0 x 6	Aerobic
0 x 7	Aerobic
0 x 8	Anaerobic
0 x 9	Anaerobic
0 x A	Stretch
0 x B	Stretch

Mechanical/Ergonomic Settings Related Bytes (Bytes 2–7)

If the HEADER COMMAND=0x02 (Learn Mode) then the Level settings are decoded from Bytes 3 to 7. Extract the bits as depicted in earlier. If the setting is designated as alpha by looking at the corresponding bit in Byte 2, then add 0x40 to the extracted bits. The client's CWR file is examined and the exercise record corresponding to the current exercise name (or lookup number) is accessed. The seatx fields are then updated accordingly.

If the HEADER COMMAND=0x00 (Normal Mode) then Bytes 2–7 are ignored.

Intensity Setting Byte (Byte 8)

The numeric value of the intensity byte is stored in the intensity field of the new CWH record.

Level Setting Format Byte (Byte 9 Bit 7)

The level setting format is noted by the program as Bit 7 or Byte 9 set=Alpha, bit cleared=Numeric

Level Setting Byte (Byte 9 Bits 6–0)

The numeric data is extracted from the byte. If the Level Setting Format designates it as a Alpha character, then the data is treated as an ASCII byte (the 8th bit is 0 anyway) and stored to the level field of the new CWH record. If the format specifies the data value as numeric, the data must be converted to characters then stored in the level field of the new CWH record.

Speed Units Code Setting Byte (Byte 10 Bits 7–5)

The Speed Units are not stored in the CWH file and thus ignored

Incline Setting Byte (Byte 10 Bits 4–0)

The incline data is extracted from Bits 4 to 0 from Byte 10 and stored directly to the incline field of the new CWH record.

Speed Decimal Indicator Byte (Byte 11 Bit 7)

The presence of a decimal point in the speed value is noted by examining Bit 7 of Bytel 1. A 1=decimal point present, 0=no decimal point.

Speed Setting Bytes (Byte 11 Bits 6–0 & Byte 12)

The speed data is extracted from Byte 11 Bits 6 to 0 (MSB) and Byte 12 (LSB). If the Speed Decimal Indicator indicates a decimal point, the extracted value is divided by 10 and then stored in the speed field of the new CWH record. If there is no decimal point indicated, the extracted value is stored without dividing first.

Calories Setting Byte (Byte 13)

The calories value is extracted from Byte 13 and first multiplied by 10 before storing directly to the calories field of the new CWH record.

Distance Unit Code Setting Byte (Byte 14 Bits 7–6)

The Distance Units are not stored in the CWH file and thus ignored

Distance Decimal Indicator Byte (Byte 14 Bit 5)

The presence of a decimal point in the distance value is noted by examining Bit 5 of Byte 14. A 1=decimal point present, 0=no decimal point.

Distance Setting Bytes (Byte 14 Bits 4–0 & Byte 15)

The distance data is extracted from Byte 14 Bits 4 to 0 (MSB) and Byte 15 (LSB). If the Distance Decimal Indicator indicates a decimal point, the extracted value is divided by 10 and then stored in the distance field of the new CWH record. If there is no decimal point indicated, the extracted value is stored without dividing first.

Duration Setting Byte (Byte 16)

The duration setting is extracted from Byte 16 and stored directly to the duration field of the new CWH record.

Duration Unit Code Setting Byte (Byte 17 Bits 7–4)

The Duration Units are not stored in the CWH file and thus ignored

Lookup Table # Bytes (Bytes 18–19)

The Lookup Table code value is extracted from Byte 18 (LSB) and Byte 19 (MSB). The program opens the lookups.dbf file in the appropriate subdirectory and searches the database for a match between the table_no field and the extracted lookup code value. The matching lookups.dbf record's exercise field value is then copied to the exercise field in the new CWH record. The lookups.dbf file is then closed.

ex_ set Field Setting

The ex_set of the new CWH record is set by extracting the value from the lower three bits of the upper nibble of the Exercise Group Byte.

Cleanup

The new CWH exercise record is now complete and appended to the CWH file. A new, blank CWH record is made with the ex_date, pulse and weight fields filled first.

Anaerobic Exercise w/Embedded Exercise Name

If the lower nibble of the exercise block command byte= 0x8 then the following bytes belong to an Anaerobic Exercise w/Embedded Exercise Name.

If the 7th byte's most significant bit (bit—7) is cleared (0) then the completed field of the new CWH record is loaded with a logical true.

For all cases of the Aerobic Exercise w/Embedded Name command byte, the following extraction and recording procedure is performed.

NOTE: The following procedure description refers to specific byte numbers. For clarification, the bytes in the

described exercise block start with the arbitrary number 1. The actual byte number in the Received DTA Data Package Buffer will depend upon its actual position amongst the other exercise blocks.

BYTE 1	Exercise Group	
BYTE 2	BIT 7:	Spare
	BIT 6:	Setting 7 1 = ALPHA 0 = NUMERIC
	BIT 5:	Setting 6 1 = ALPHA 0 = NUMERIC
	BIT 4:	Setting 5 1 = ALPHA 0 = NUMERIC
	BIT 3:	Setting 4 1 = ALPHA 0 = NUMERIC
	BIT 2:	Setting 3 1 = ALPHA 0 = NUMERIC
	BIT 1:	Setting 2 1 = ALPHA 0 = NUMERIC
	BIT 0:	Setting 1 1 = ALPHA 0 = NUMERIC
BYTE 3	BITS 7–5:	# OF SETTINGS (0–7)
	BITS 4–0:	Setting #1 (0–32 or A–Z)
BYTE 4	333 22222	Setting #2 & #3
BYTE 5	5 44444 33	Setting #3, #4 & #5
BYTE 6	6666 5555	Setting #5 & #6
BYTE 7	XX 77777 6	Setting #6 & #7
	BIT 6:	Spare
	BIT 7:	Completed Status 0 = completed as prescribed 1 = modified
BYTE 8	BIT 7:	0 = POUNDS 1 = PLATES
	BIT 6:	1 = Decimal in pounds/Plate O= No decimal
	BITS 5–0:	POUNDS/PLATES MSB
BYTE 9	BITS 7–0	POUNDS/PLATES (0–16383 (1638.3)) LSB
BYTE 10	BITS 7–4	Duration type 0 = seconds 1 = minutes 2 = hours
	BITS 3–0	Spare
BYTE 11	BITS 7–0	DURATION (0–255)
BYTE 12	BITS 7–0	REPS (0–255)
BYTE 13	BITS 7–4:	SETS (0–15)
	BITS 3–0	# CHARACTERS IN EQUIPMENT NAME (1–16) +1
BYTE 14	222 11111	
BYTE 15	4 33333 22	
BYTE 16	5555 4444	
BYTE 17	77 66666 5	
BYTE 18	88888 777	
BYTE 19	AAA 99999	
BYTE 20	C BBBBB AA	
BYTE 21	DDDD CCCC	
BYTE 22	FF EEEEE D	
BYTE 23	GGGGG FFF	

Exercise Group Byte (Byte 1)

The Exercise Group byte is decoded to determine what exercise group the block represents. The lower nibble is used to determine the exercise group. The resultant decoded value is stored to the ex_group field of the new CWH record as follows:

Command Byte Lower Nibble	ex_group
0x6	Aerobic
0x7	Aerobic
0x8	Anaerobic
0x9	Anaerobic
0xA	Stretch
0xB	Stretch

Mechanical/Ergonomic Settings Related Bytes (Bytes 2–7)

If the HEADER COMMAND=0x02 (Learn Mode) then the Level settings are decoded from Bytes 3 to 7. Extract the bits as depicted earlier. If the setting is designated as alpha by looking at the corresponding bit in Byte 2, then add 0x40 to the extracted bits. The client's CWR file is examined and the exercise record corresponding to the current exercise name (or lookup number) is accessed. The seatx fields are then updated accordingly.

If the HEADER COMMAND=0x0 (Normal Mode) then Bytes 2–7 are ignored.

Pounds/Plates Type Setting Byte (Byte 8 Bit 7)

The Pounds/Plates Type Setting is not stored in the CWH and thus ignored

Pounds/Plates Decimal Indicator Byte (Byte 8 Bit 6)

The presence of a decimal point in the pounds value is noted by examining Bit 6 of Byte 8. A 1=decimal point present, 0=no decimal point.

Pounds/Plates Setting Bytes (Byte 8 Bits 5–0 & Byte 9)

The pounds data is extracted from Byte 8 Bits 5 to 0 (MSB) and Byte 9 (LSB). If the Pounds/Plates Decimal Indicator indicates a decimal point, the extracted value is divided by 10 and then stored in the pounds field of the new CWH record. If there is no decimal point indicated, the extracted value is stored without dividing first.

Duration Unit Code Setting Byte (Byte 10 Bits 7–4)

The Duration Units are not stored in the CWH file and thus ignored

Duration Setting Byte (Byte 11)

The duration setting is extracted from Byte 11 and stored directly to the duration field of the new CWH record.

Repetitions Setting Byte (Byte 12)

The repetitions value is extracted from Byte 12 and stored directly to the reps field of the new CWH record.

Sets setting Byte (Byte 13 Bits 7–4)

The sets value is extracted from Byte 12 and stored directly to the reps field of the new CWH record.

# of Characters In Exercise Name Setting Byte (Byte 13 Bits 3–0)

The number of Characters value is used in the following section to extract the exercise name from the packed bytes.

Exercise Name Packed Bytes (Bytes 14–23)

In a similar fashion as the packing of the exercise name, the embedded exercise name characters are extracted and unpacked.

The ASCII characters have been trimmed to 5 bits before packing and thus when extracted, the upper 3 bits are restored by adding 0x40 to each extracted character.

NOTE: The ASCII space character, 0x20, is packed as 0. When an extracted 5-bit character=0, add 0x20 instead of 0x40.

The program concatenates the exercise name string one character at a time. When the number of characters equals the Number of Characters value extracted, the string is stored to the exercise field of the new CWH record.

BYTE 14	222 11111	1st Character, partial 2nd
BYTE 15	4 33333 22	remaining 2nd, complete 3rd & partial 4th
BYTE 16	5555 4444	remaining 4th, partial 5th
BYTE 17	77 66666 5	remaining 5th, complete 6th, partial 7th
BYTE 18	88888 777	remaining 7th, complete 8th
BYTE 19	AAA 99999	complete 9th, partial 10th
BYTE 20	C BBBBB AA	remaining 10th, complete 11th, partial 12th
BYTE 21	DDDD CCCC	remaining 12th, partial 13th
BYTE 22	FF EEEEE D	remaining 13th, complete 14th, partial 15th
BYTE 23	GGGGG FFF	remaining 15th, complete 16th

ex_set Field Setting

The ex_set of the new CWH record is set by extracting the value from the lower three bits of the upper nibble of the Exercise Group Byte.

Cleanup

The new CWH exercise record is now complete and appended to the CWH file. A new, blank CWH record is made with the ex_date, pulse and weight fields filled first.

Anaerobic Exercise with Lookup Table Code Exercise Name

If the lower nibble of the exercise block command byte= 0x9 then the following bytes belong to an Anaerobic Exercise with Lookup Table Code Exercise Name

If the 7th byte's most significant bit (bit—7) is cleared (0) then the completed field of the new CWH record is loaded with a logical true.

For all cases of the Aerobic Exercise with Lookup Table Code Exercise Name command byte, the following extraction and recording procedure is performed.

NOTE: The following procedure description refers to specific byte numbers. For clarification, the bytes in the described exercise block start with the arbitrary number 1. The actual byte number in the Received DTA Data Package Buffer will depend upon its actual position amongst the other exercise blocks.

BYTE 1	Exercise Group	
BYTE 2	BIT 7:	Spare
	BIT 6:	Setting 7 1 = ALPHA 0 = NUMERIC
	BIT 5:	Seyting 6 1 = ALPHA 0 = NUMERIC
	BIT 4:	Setting 5 1 = ALPHA 0 = NUMERIC
	BIT 3:	Setting 4 1 = ALPHA 0 = NUMERIC
	BIT 2:	Setting 3 1 = ALPHA 0 = NUMERIC
	BIT 1:	Setting 2 1 = ALPHA 0 = NUMERIC
	BIT 0:	Setting 1 1 = ALPHA 0 = NUMERIC
BYTE 3	BITS 7–5:	# OF SETTINGS (0–7)
	BITS 4–0:	Setting #1 (0–32 or A–Z)
BYTE 4	333 22222	Setting #2 & #3
BYTE 5	5 44444 33	Setting #3, #4 & #5
BYTE 6	6666 5555	Setting #5 & #6
BYTE 7	XX 77777 6	Setting #6 & #7
	BIT 6:	Spare
	BIT 7:	Completed Status 0 = completed as prescribed 1 = modified
BYTE 8	BIT 7:	0 = POUNDS 1 = PLATES
	BIT 6:	1 = Decimal in pounds/Plate 0 = No decimal
	BITS 5–0:	POUNDS/PLATES MSB
BYTE 9	BITS 7–0	POUNDS/PLATES (0–16383 (1638.3)) LSB
BYTE 10	BITS7–0	REPS (0–255)
BYTE 11	BITS 7–4:	SETS (0–15)
	BITS 3–0	Duration type 0 = seconds 1 = minutes 2 = hours
BYTE 12	BITS 7–0	DURATION (0–255)
BYTE 13	LOOKUP TABLE # LSB	
BYTE 14	LOOKUP TABLE # MSB	

Exercise Group Byte (Byte 1)

The Exercise Group byte is decoded to determine what exercise group the block represents. The lower nibble is used to determine the exercise group. The resultant decoded value is stored to the ex_group field of the new CWH record as follows:

Command Byte Lower Nibble	ex_group
0 × 6	Aerobic
0 × 7	Aerobic
0 × 8	Anaerobic
0 × 9	Anaerobic
0 × A	Stretch
0 × B	Stretch

Mechanical/Ergonomic Settings Related Bytes (Bytes 2–7)

These bytes are not stored in the CWH and thus ignored.

Pounds/Plates Type Setting Byte (Byte 8 Bit 7)

The Pounds/Plates Type Setting is not stored in the CWH and thus ignored

Pounds/Plates Decimal Indicator Byte (Byte 8 Bit 6)

The presence of a decimal point in the pounds value is noted by examining Bit 6 of Byte 8. A 1=decimal point present, 0=no decimal point.

Pounds/Plates Setting Bytes (Byte 8 Bits 5–0 & Byte 9)

The pounds data is extracted from Byte 8 Bits 5 to 0 (MSB) and Byte 9 (LSB). If the Pounds/Plates Decimal Indicator indicates a decimal point, the extracted value is divided by 10 and then stored in the pounds field of the new CWH record. If there is no decimal point indicated, the extracted value is stored without dividing first.

Repetitions Setting Byte (Byte 10)

The repetitions value is extracted from Byte 10 and stored directly to the reps field of the new CWH record.

Sets setting Byte (Byte 1 Bits 7–4)

The sets value is extracted from Byte 11 Bits 7 to 4 and stored directly to the sets field of the new CWH record.

Duration Unit Code Setting Byte (Byte 11 Bits 3–0)

The Duration Units are not stored in the CWH file and thus ignored

Duration Setting Byte (Byte 12)

The duration setting is extracted from Byte 11 and stored directly to the duration field of the new CWH record.

Lookup Table # Bytes (Bytes 13–14)

The Lookup Table code value is extracted from Byte 13 (LSB) and Byte 14 (MSB).The program opens the lookups.dbf file in the appropriate subdirectory and searches the database for a match between the table_no field and the extracted lookup code value. The matching lookups.dbf record's exercise field value is then copied to the exercise field in the new CWH record. The lookups.dbf file is then closed.

ex_set Field Setting

The ex_set of the new CWH record is set by extracting the value from the lower three bits of the upper nibble of the Exercise Group Byte.

Cleanup

The new CWH exercise record is now complete and appended to the CWH file. A new, blank CWH record is made with the ex_date, pulse and weight fields filled first.

Stretch Exercise w/Embedded Exercise Name

If the lower nibble of the exercise block command byte= 0xA then the following bytes belong to an Stretch Exercise w/Embedded Exercise Name.

If the 3rd byte's least significant bit (bit 0) is cleared (0) then the completed field of the new CWH record is loaded with a logical true.

For all cases of the Stretch Exercise w/Embedded Name command byte, the following extraction and recording procedure is performed.

NOTE: The following procedure description refers to specific byte numbers. For clarification, the bytes in the described exercise block start with the arbitrary number 1. The actual byte number in the Received DTA Data Package Buffer will depend upon its actual position amongst the other exercise blocks.

BYTE 1	Exercise Group	
BYTE 2	BITS 7–0:	DURATION (0–255)
BYTE 3	BITS 7–4:	Duration type 0 = seconds 1 =minutes 2=hours
	BITS 3–1:	Spare
	BIT 0:	Completed Status 0 = completed as prescribed 1 = modified
BYTE 4	BIT 7–0:	REPS (0–255)
BYTE 5	BITS 7–4:	SETS (0–15)
	BITS 3–0	# CHARACTERS IN EQUIPMENT NAME (1–16) +1
BYTE 6	222 11111	a
BYTE 7	4 33333 22	b
BYTE 8	5555 4444	c
BYTE 9	77 66666 5	d
BYTE 10	88888 777	e
BYTE 11	AAA 99999	a
BYTE 12	C BBBBB AA	b

-continued

BYTE 13	DDDD CCCC	c
BYTE 14	FF EEEEE D	d
BYTE 15	GGGGG FFF	e

Exercise Group Byte (Byte 1)

The Exercise Group byte is decoded to determine what exercise group the block represents. The lower nibble is used to determine the exercise group. The resultant decoded value is stored to the ex_group field of the new CWH record as follows:

Command Byte Lower Nibble	ex_group
0 × 6	Aerobic
0 × 7	Aerobic
0 × 8	Anaerobic
0 × 9	Anaerobic
0 × A	Stretch
0 × B	Stretch

Duration Setting Byte (Byte 2)

The duration setting is extracted from Byte 2 and stored directly to the duration field of the new CWH record.

Duration Unit Code Setting Byte (Byte 3 Bits 7–4)

The Duration Units are not stored in the CWH file and thus ignored

Repetitions Setting Byte (Byte 4)

The repetitions value is extracted from Byte 4 and stored directly to the reps field of the new CWH record.

Sets setting Byte (Byte 5 Bits 7–4)

The sets value is extracted from Byte 5 Bits 7 to 4 and stored directly to the sets field of the new CWH record.

# of Characters In Exercise Name Setting Byte (Byte 5 Bits 3–0)

The number of Characters value is used in the following section to extract the exercise name from the packed bytes.

Exercise Name Packed Bytes (Bytes 6–15)

In a similar fashion as the packing of the exercise name, the embedded exercise name characters are extracted and unpacked.

The ASCII characters have been trimmed to 5 bits before packing and thus when extracted, the upper 3 bits are restored by adding 0x40 to each extracted character.

NOTE: The ASCII space character, 0x20, is packed as 0. When an extracted 5-bit character=0, add 0x20 instead of 0x40.

The program concatenates the exercise name string one character at a time. When the number of characters equals the Number of Characters value extracted, the string is stored to the exercise field of the new CWH record.

BYTE 6	222 11111	1st Character, partial 2nd
BYTE 7	4 33333 22	remaining 2nd, complete 3rd & partial 4th
BYTE 8	5555 4444	remaining 4th, partial 5th
BYTE 9	77 66666 5	remaining 5th, complete 6th, partial 7th
BYTE 10	88888 777	remaining 7th, complete 8th
BYTE 11	AAA 99999	complete 9th, partial 10th
BYTE 12	C BBBBB AA	remaining 10th, complete 11th, partial 12th
BYTE 13	DDDD CCCC	remaining 12th, partial 13th
BYTE 14	FF EEEEE D	remaining 13th, complete 14th, partial 15th
BYTE 15	GGGGG FFF	remaining 15th, complete 16th

ex_set field Setting

The ex_set of the new CWH record is set by extracting the value from the lower three bits of the upper nibble of the Exercise Group Byte.

Cleanup

The new CWH exercise record is now complete and appended to the CWH file. A new, blank CWH record is made with the ex_date, pulse and weight fields filled first.

Stretch Exercise with Lookup Table Code Exercise Name

If the lower nibble of the exercise block command byte= 0xB then the following bytes belong to an Stretch Exercise with Lookup Table Code Exercise Name.

If the 3rd byte's least significant bit (bit 0) is cleared (0) then the completed field of the new CWH record is loaded with a logical true.

For all cases of the Stretch Exercise with Lookup Table Code Exercise Name exercise group byte, the following extraction and recording procedure is performed.

NOTE: The following procedure description refers to specific byte numbers. For clarification, the bytes in the described exercise block start with the arbitrary number 1. The actual byte number in the Received DTA Data Package Buffer will depend upon its actual position amongst the other exercise blocks.

BYTE 1	Exercise Group
BYTE 2	BITS 7–0: DURATION (0–255)
BYTE 3	BITS 7–4: Duration type 0 = seconds 1 = minutes 2 = hours
	BITS 3–1: Spare
	BIT 0: Completed Status 0 = completed as prescribed 1 = modified
BYTE 4	BIT 7–0: REPS (0–255)
BYTE 5	BITS 7–4: SETS (0–15)
	BITS 3–0 spare
BYTE 6	Lookup Table # LSB
BYTE 7	Lookup Table # MSB

Exercise Group Byte (Byte 1)

The Exercise Group byte is decoded to determine what exercise group the block represents. The lower nibble is used to determine the exercise group. The resultant decoded value is stored to the ex_group field of the new CWH record as follows:

Command Byte Lower Nibble	ex_group
0 × 6	Aerobic
0 × 7	Aerobic
0 × 8	Anaerobic
0 × 9	Anaerobic
0 × A	Stretch
0 × B	Stretch

Duration Setting Byte (Byte 2)

The duration setting is extracted from Byte 2 and stored directly to the duration field of the new CWH record.

Duration Unit Code Sting Byte (Byte 3 Bits 7–4)

The Duration Units are not stored in the CWH file and thus ignored

Repetitions Setting Byte (Byte 4)

The repetitions value is extracted from Byte 4 and stored directly to the reps field of the new CWH record.

Sets setting Byte (Byte 5 Bits 7–4)

The sets value is extracted from Byte 5 Bits 7 to 4 and stored directly to the sets field of the new CWH record.

Lookup Table # Bytes (Bytes 6–7)

The Lookup Table code value is extracted from Byte 6 (LSB) and Byte 7 (MSB). The program opens the lookups-.dbf file in the appropriate subdirectory and searches the database for a match between the table_no field and the extracted lookup code value. The matching lookups.dbf record's exercise field value is then copied to the exercise field in the new CWH record. The lookups.dbf file is then closed.

ex_set Field Setting

The ex_set of the new CWH record is set by extracting the value from the lower three bits of the upper nibble of the Exercise Group Byte.

Cleanup

The new CWH exercise record is now complete and appended to the CWH file. A new, blank CWH record is made with the ex_date, pulse and weight fields filled first.

I claim:

1. An exercise guidance system for guiding an individual through a workout schedule using an adaptive workout routine that includes a series of selected exercises and a progressive series of intermediate performance goals for each of the exercises, each of the progressive series of intermediate performance goals being calculated, for each exercise, based upon the individual's current performance of the exercise, a previous intermediate performance goal for the exercise and a selected final performance goal for the exercise using a preselected, but variable, set of calculation parameters, said system comprising:

a portable data recorder/display unit having

data input means for manually entering data into said portable data recorder/display unit representative of an individual's current performance of an exercise,

recorder memory means capable of storing data for later retrieval,

display means capable for displaying data,

input/output port means for communicating data between said recorder memory means and an external device, and

data controller means for storing in and retrieving from said recorder memory means data entered from the data input means and displaying from said recorder memory means selected data on said display means, said controller means also capable of sending data stored in said recorder memory means to, and receiving data from, said external device through said input/output port means; and

a separate base programming unit having

processing means, external from said portable data recorder/display unit, having memory means containing a control program and stored data representing, for each of the exercises to be performed by the individual, and individual's current performance of the exercise, a preselected initial intermediate performance goal and a final performance goal, a preselected plurality of calculation parameters, said processing means calculating a subsequent intermediate performance goal using said individual's current performance of the exercise, a previous intermediate performance goal for the exercise and said final performance goal for the exercise using said plurality of calculation parameters, wherein said processing means communicating said subsequent performance goal to said portable data recorder/display unit for display thereon for guidance of the individual in performing the exercise for each exercise to be performed by the individual, and

a docking station unit adapted to mate electrically with said input/output port means of said portable data recorder/display unit for communicating data between said portable data recorder/display unit and said processing means of said base programming unit.

2. The exercise guidance system as in claim 1 wherein said base programming unit further comprises:

data input means for manually entering modified data representing said initial and said subsequent intermediate performance and said final performance goals and said plurality of calculation parameters in said memory means of said processing means.

3. The exercise guidance system as in claim 1 wherein said base programming unit further comprises:

display means capable of visually displaying selected data from said memory means.

4. The exercise guidance system as in claim 1 wherein said programming unit further comprises:

docking station unit having a shape adapted to receive and retain releasably said portable data recorder/display unit.

5. The exercise guidance system as in claim 1 wherein said base programming unit further comprises:

connection means for connecting said processing means of said base programming unit to said docking station unit for communicating data therebetween.

6. The exercise guidance system as in claim 5 wherein said connection means comprises a cable means.

7. The exercise guidance system as in claim 1 wherein said memory means of said processing means of said base programming unit is further operable for storing an historical database of data representative of said individual's current performance of the exercise for evaluation with respect to some further exercise activity.

8. The exercise guidance system as in claim 1 wherein said processing means of said base programming unit includes a programmed microcomputer.

9. The exercise guidance system as in claim 1 wherein said display means of said portable data recorder/display unit includes an alphanumeric display means.

10. The exercise guidance system as in claim 9 wherein said alphanumeric display means comprises a liquid crystal display.

11. The exercise guidance system as in claim 10 wherein said memory means of said processing means of said base programming unit is further operable for storing an historical database of data representative of said individual's current performance of the exercise for evaluation with respect to some further exercise activity.

12. The exercise guidance system as in claim 10 wherein said processing means of said base programming unit includes a programmed microcomputer.

13. The exercise guidance system as in claim 10 wherein said controller means of said portable data recorder/display unit is adapted to store and retrieve separately data entered by a plurality of individuals.

14. The exercise guidance system as in claim 10 wherein said processing means of said base programming unit and stored data in said memory means of said processing means are adapted to calculate subsequent intermediate performance goals for a plurality of individuals.

15. The exercise guidance system as in claim 10 wherein said processing means of said base programming unit and stored data in said memory means of said processing means are adapted to calculate subsequent intermediate performance goals for a plurality of exercises.

16. The exercise guidance system as in claim 1 wherein said controller means of said portable data recorder/display unit is adapted to store and retrieve separately data entered by a plurality of individuals.

17. The exercise guidance system as in claim 1 wherein said processing means of said base programming unit and stored data in said memory means of said processing means are adapted to calculate subsequent intermediate performance goals for a plurality of individuals.

18. The exercise guidance system as in claim 1 wherein said processing means of said base programming unit and

stored data in said memory means of said processing means are adapted to calculate subsequent intermediate performance goals for a plurality of exercises.

19. An exercise guidance system for guiding an individual through a workout schedule using an adaptive workout routine that includes a series of selected exercises and a progressive series of intermediate performance goals for each of the exercises, each of the progressive series of intermediate performance goals being calculated, for each exercise, based upon the individual's current performance of the exercise, a previous intermediate performance goal for the exercise and a selected final performance goal for the exercise using a preselected, but variable, set of calculation parameters, said system comprising:

a portable data recorder/display unit having

data input means for manually entering data into said portable data recorder/display unit representative of an individual's current performance of an exercise,

recorder memory means capable of storing data for later retrieval,

display means capable for displaying data,

input/output port means for communicating data between said recorder memory means and an external device, and

data controller means for storing in and retrieving from said recorder memory means data entered from the data input means and displaying from said recorder memory means selected data on said display means, said controller means also capable of sending data stored in said recorder memory means to, and receiving data from, said external device through said input/output port means; and

a separate base programming unit having

processing means, external from said portable data recorder/display unit, having memory means containing a control program and stored data representing, for each of the exercises to be performed by the individual, and individual's current performance of the exercise, a preselected initial intermediate performance goal and a final performance goal, a preselected plurality of calculation parameters, said processing means calculating a subsequent intermediate performance goal using said individual's current performance of the exercise, a previous intermediate performance goal for the exercise and said final performance goal for the exercise using said plurality of calculation parameters, wherein said processing means communicating said subsequent performance goal to said portable data recorder/display unit for display thereon for guidance of the individual in performing the exercise for each exercise to be performed by the individual,

data input means for manually entering modified data representing said initial and said subsequent intermediate performance and said final performance goals and said plurality of calculation parameters in said memory means of said processing means;

display means capable of visually displaying selected data from said memory means; and

a docking station unit adapted to mate electrically with said input/output port means of said portable data recorder/display unit for communicating data between said portable data recorder/display unit and said processing means of said base programming unit, and having a shape adapted to receive and retain releasably said portable data recorder/display unit.

20. The exercise guidance system as in claim 19 wherein said base programming unit further comprises:

connection means for connecting said processing means of said base programming unit to said docking station unit for communicating data therebetween.

21. The exercise guidance system as in claim 20 wherein said connection means comprises a cable means.

22. The exercise guidance system as in claim 20 wherein said display means of said portable data recorder/display unit includes an alphanumeric display means.

23. The exercise guidance system as in claim 22 wherein said alphanumeric display means comprises a liquid crystal display.

* * * * *